

ওয়ার্ডপ্ৰেস প্লাগিন ডেভেলপমেন্ট



Version : 1.0

Written By

মো : লিটন আৰেফিন

(Beginner's to Advance)

[It Is Free & Next Versions Will Also Be Free]

Published By : WordPress Group

<http://goo.gl/YwsDG>

ওয়ার্ডপ্ৰেস প্লাগিন ডেভেলপমেন্ট (Beginner's to Advance)

মোঃ লিটন আরেফিন

<https://www.facebook.com/litonce09>

প্রথম প্রকাশ
১০ সেপ্টেম্বর ২০১২

প্রকাশক
ওয়ার্ডপ্রেস গ্রুপ বাংলাদেশ
“<https://www.facebook.com/groups/Wordpress2Smashing/>”

ওয়ার্ডপ্রেস গ্রুপ ওয়েবসাইট
“<https://www.wpbangla.com>”

লেখক: মোঃ লিটন আরেফিন
“<https://www.facebook.com/litonic09>”
“www.liton-online.com”

প্রচ্ছদ:
জামিল হোসেন সিজান

স্পন্সর:
“Shirsendu Shekhar Biswas”
“<https://www.facebook.com/dip.shekhar>”

Caution
This Book Not For sale
(CopyRight @ 2012 By ওয়ার্ডপ্রেস গ্রুপ বাংলাদেশ)

আমার সম্পর্কে

আমার নাম মোঃ লিটন আরেফিন (<https://www.facebook.com/litonice09>)। রাজশাহী বিশ্ববিদ্যালয় থেকে ইনফরমেশন এন্ড কমিউনিকেশন ইঞ্জিনিয়ারিং বিভাগে পড়াশোনা করছি, অনার্স শেষ এখন মাস্টার্স করছি। আমার ব্যক্তিগত ব্লগ “[লিটন- অনলাইন.কম](http://www.liton-online.com)” (www.liton-online.com) এ টাইম দেই যতটা পারি, সাথে টুকটাক ফ্রিল্যান্সিং করি। আমাদের তিন সদস্যের গ্রুপ “[বুরহান](http://burhanbd.com/)”(http://burhanbd.com/) - এ কাজ করি এভাবেই সময় চলে যায় আমার। আমার প্লাগিন খুব বেশি ডেভেলপ করা হয়নি তবুও জামিল ভাই আমাকে এই বইটি লেখার কাজ দিয়েছেন। আমি আমার স্বপ্ন জ্ঞানে চেষ্টা করব আমার সেরাটা দেবার। জামিল- (“<https://www.facebook.com/zamil.hossainsezan32>)” ভাই আমাকে ওয়ার্ডপ্রেস গ্রুপ এর- (“<https://www.facebook.com/groups/Wordpress2Smashing/>”) পক্ষ থেকে এই বইটি লেখার কাজ অনেক দিন আগেই দিয়েছিলেন। কিন্তু সময় করে উঠতে পারছিলাম না। গত মাসেই আমার বইটি হস্তান্তর করার কথা ছিল কিন্তু পারিনি।

এই বইটির স্বর্স্বত্ব আমার। কেউ এই বইটি বিক্রি করা বা নিজের নামে চালানোর চেষ্টা করবেন না। তবে যত বেশি পারেন সবার সাথে শেয়ার করবেন। আমরা শুধু ওয়েবের জগতে ডাটা এন্ট্রি এক্সপার্ট বা ক্লিকার হয়ে থাকতে চাইনা। ওয়েবের এডভান্স লেভেলে আপনাকে স্বাগতম।

মোঃ লিটন আরেফিন

সূচিপত্র

অধ্যায়- একঃ সূচনা	পেজ নম্বর
কিছু কথা	৭
বইটি কাদের জন্য?	৭
প্রয়োজনীয় টুলস	৮
অধ্যায়- দুইঃ প্লাগিনের গঠন	
প্লাগিন কি?	৯
প্লাগিন ওয়ার্ডপ্রেসের সাথে কিভাবে কাজ করে?	৯
কখন আপনার প্লাগিন লোড হয়?	১১
ওয়ার্ডপ্রেস প্লাগিন ডেভেলপ করার ৭টি কারণ	১৩
প্লাগিনের ফোল্ডার স্ট্রাকচার কেমন হবে	১৪
প্লাগিন হেডার	১৫
প্লাগিন পাথ	১৮
লোকাল সার্ভার পাথঃ	১৮
URL পাথ	১৯
অধ্যায়- তিনঃ হুকস	
ওয়ার্ডপ্রেস হুক(Hook) কি?	২২
ফিল্টার হুক (Filter Hooks)	২২
এ্যাকশন হুক (Action Hooks)	২৭
অধ্যায়- চারঃ প্লাগিন কোডিং স্ট্যান্ডার্ড	
প্লাগিন এক্টিভেশন/ ডিএক্টিভেশন ফাংশন	৩২
uninstall.php	৩৪
Uninstall Hook	৩৫
কোডিং স্ট্যান্ডার্ড	৩৬
পিএইচপি স্ট্যান্ডার্ড	৩৭
এসকিউএল স্টেটমেন্ট	৩৭
অধ্যায়- পাঁচঃ ওয়ার্ডপ্রেসের সাথে ইন্টিগ্রেশন	
মেনু এবং সাব মেনু তৈরি করণ	৩৮
টপ লেভেল মেনু	৩৮
মেনুকে আগের মেনুর সাথে যুক্ত করা(সাব মেনু)	৪৩

উইজেট তৈরি করা ৪৫

অধ্যায়- ছয়ঃ প্লাগিন ফাংশন

কিছু বেসিক ফাংশন ৫৬

__() ফাংশন ৫৬

_e() ফাংশন ৫৬

esc_attr__() ফাংশন ৫৮

অধ্যায়- সাতঃ প্লাগিন সিকিউরিটি

সিকিউরিটি ৬১

প্লাগিনের সিকিউরিটি কি? ৬১

আপনার প্লাগিনকে কি সিকিউরিটি দেবেন? ৬১

ইউজার পারমিশনস ৬২

Nonces ৬৩

Nonce URL তৈরি ৬৪

Nonce Form তৈরি ৬৫

ডাটাবেস ৭৩

অধ্যায়- আটঃ লোকালাইজেশন এবং Set t i ngs API

লোকালাইজেশন ৭৬

সেটিংস API ৭৬

Settings API ফাংশন ৭৭

সেটিংস ডিফাইন সেকশন ৮১

CSS/JS ফাইল যুক্ত করা ৯১

সর্টকোড ৯৩

অধ্যায়- নয়ঃ প্লাগিন উদাহরণ

ওয়ার্ডপ্রেসের ডেট/টাইম/কমেন্ট নাম্বার বাংলা করার প্লাগিন ১০০

স্লাইডার প্লাগিন ১০১

অধ্যায়- একঃ সূচনা

কিছু কথা

ওয়ার্ডপ্রেসের প্রতি রয়েছে অনেক ভালবাসা। ওয়ার্ডপ্রেস সম্পর্কে নতুন করে কিছু বলার নেই। যেকোন ওয়ার্ডপ্রেস ব্যবহারকারীর জন্য প্লাগিন একটি গুরুত্বপূর্ণ বিষয়। প্লাগিন আসলে কি? ওয়ার্ডপ্রেস প্লাগিন আমাদেরকে আমাদের ব্লগ/ ওয়েবসাইটের ফাংশনালিটি বৃদ্ধি করতে সাহায্য করে, সামান্য অথবা কোন রকম প্রোগ্রামিং জ্ঞান ছাড়াই। ওয়ার্ডপ্রেস প্লাগিনের নির্দিষ্ট কোন আকার এবং আকৃতি নেই, একটি প্লাগিন যেকোন কিছুই করতে পারে ওয়ার্ডপ্রেস ব্লগ/ ওয়েবসাইটের।

আমার প্লাগিনের সংখ্যা খুব কম। কিছু পাবলিশ করা হয়নি। আমি ওয়ার্ডপ্রেসের কিছু লিমিটেশন নিয়ে কাজ করছি এবং এগুলো নিয়েই মূলত কাজ করছি কিভাবে আরো এক্সটেন্ড করা যায়। যাই হোক আমি এখনও শিক্ষানবিশ হিসেবে আছি।

বইটি কাদের জন্য?

আপনি যদি এই বইটি ডাউনলোড করে থাকেন তাহলে আমি ধরেই নিচ্ছি আপনি একজন ওয়েব ডেভেলপার এবং সাথে কিছু প্রোগ্রামিং জ্ঞান আছে। প্লাগিন ডেভেলপমেন্টের উদ্দেশ্য হচ্ছে আপনি প্লাগিন ডেভেলপ করবেন আপনার ক্লায়েন্ট/ আপনি যেভাবে চান ঠিক সেভাবে।

যেকোন ধরনের ওয়েবসাইট তৈরীতে ওয়ার্ডপ্রেসের ব্যতিক্রম প্ল্যাটফর্ম ইতোমধ্যেই প্রমাণিত।

এই বইটি ফ্রিল্যান্সারদের জন্যেও যারা তাদের স্কিল পোর্টফোলিও, এবং ওয়ার্ডপ্রেসের ফাংশনালিটি বুঝে কাজ করেন।

প্রয়োজনীয় টুলস

- [Notepad ++/\(http://notepad-plus-plus.org/download/ \)](http://notepad-plus-plus.org/download/) Dreamwaver
- [Mozilla Firefox \(http://www.mozilla.org/en-US/firefox/new/\)](http://www.mozilla.org/en-US/firefox/new/) /[Google Chrome](#)
- Firebug Extension (Search For both Firefox/ Chrome as you want)
- Web Developer Extension (Search For both Firefox/ Chrome as you want)
- Xampp with Wordpress Installation
- FTP Client. Example- Filezilla.

ওয়ার্ডপ্রেস কিভাবে ইনস্টল করতে হয় তা নতুন করে দেখানোর কিছু নেই তবুও যারা নতুন তারা [এখানে দেখুন](http://goo.gl/WJTp9) (<http://goo.gl/WJTp9>) ।

কিভাবে প্লাগিন ইনস্টল করতে হয় তা [এখানে দেখুন](http://goo.gl/pE8AD) (<http://goo.gl/pE8AD>) ।

অধ্যায়- দুইঃ প্লাগিনের গঠন

প্লাগিন কি?

ওয়ার্ডপ্রেসের প্লাগিন হল একটি PHP স্ক্রিপ্ট যা ওয়ার্ডপ্রেসের নিজস্ব ফাংশনালিটি পরিবর্তন করে অথবা পরিবর্ধন করে। মূলতঃ একটি ফিচার বা কিছু ফিচার আমাদের সাইটে যোগ করার জন্য আমরা প্লাগিন ইনস্টল দেই। একটি প্লাগিনের কাজের কোন লিমিট নেই সে কোন কাজগুলো করতে পারবে, কারন প্লাগিন ডাউনলোডের কোন নির্দিষ্ট সাইজ নেই।

প্লাগিন ওয়ার্ডপ্রেসের সাথে কিভাবে কাজ করে?

ওয়ার্ডপ্রেসে অনেক ফিচার আছে, বিভিন্ন API(Application Programming Interface)-এর জন্য যেগুলো বিভিন্নভাবে কাজ করে ওয়ার্ডপ্রেসে। নিচে সেগুলো আলোচনা করা হলঃ

- **প্লাগিনঃ** একসেট হুক(hooks) যেগুলো ওয়ার্ডপ্রেসের নির্দিষ্ট ফাংশনে এক্সেস করে এগুলোকে এনাবল করে।
- **উইজেটঃ** প্লাগিন দিয়ে উইজেট তৈরি এবং পরিচালনা করা।
- **সর্টকোডঃ** আপনার প্লাগিনকে সাপোর্ট করার জন্য সর্টকোড যোগ করা। সর্টকোড হল একটি হুক যেটা একটি PHP ফাংশনকে এনাবল করে এবং আপনার পোস্ট বা পেজ এ প্রদর্শন করে। আমি পরে এ বিষয়ে বিস্তারিত বলব।
- **HTTP:** এটি আপনার প্লাগিনে HTTP রিকোয়েস্ট পাঠায়। এই API কনটেন্ট থেকে external URL অথবা Submit করা কনটেন্ট থেকে URL retrieve করে।
- **সেটিংসঃ** আপনার প্লাগিনের সেটিংস ইনসার্ট অথবা সেটিংস সেকশন। সেটিংস API এর প্রাথমিক সুবিধা হল সিকিউরিটি। সকল সেটিংস ডাটা পরিস্কার(পরিশোধিত) থাকে, সুতরাং যখন প্লাগিন সেটিংস সেভ করার সময় Cross Site Request Forgery (CSRF) এবং Cross Site Scripting(XSS) নিয়ে চিন্তার কিছু নেই।
- **Options:** প্লাগিনের সকল অপশন জমা রাখে এবং retrieve করে। এটি নতুন অপশন তৈরী করে, আগের অপশন আপডেট করে, অপশন ডিলিট করে এবং যেকোন অপশন retrieve করে।
- **Dashboard Widgets:** ড্যাশবোর্ড উইজেট তৈরি করে। ওয়ার্ডপ্রেস ড্যাশবোর্ডে automatically উইজেট প্রদর্শন এবং সকল কাস্টমাইজেশন ফিচার যেমন-মিনিমাইজ, ড্রাগ/ ড্রপ, এবং স্ক্রিন অপশন।
- **Rewrite:** আপনার প্লাগিনের জন্য কাস্টম রুলস তৈরি করা। এই API আপনার static-end-point(/custom-page), structure tags(%postname%), এবং অতিরিক্ত ফিড লিঙ্কস (/feed/json/)।

Caution
This Book Not For sale

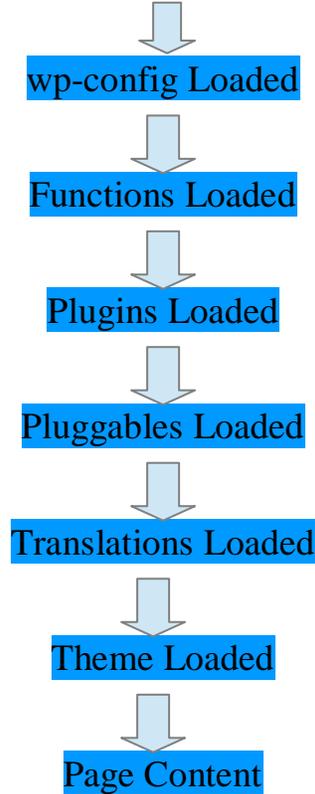
- **Transients:** কিছু অস্থায়ী অপশন (cached data) তৈরি করে আপনার প্লাগিনের জন্য। এই API টি Options API এর অনুরূপ।
- **Database:** ওয়ার্ডপ্রেসের ডাটাবেসে প্রবেশ করে। এটি আপনার প্লাগিনের ডাটাবেজে নিচের কাজগুলো সম্পাদন করে- creating, updating, deleting and retrieving ।

কখন আপনার প্লাগিন লোড হয়?

প্লাগিন তৈরি করতে গেলে এই বেসিক কাজগুলো আমাদের অবশ্যই জানতে হবে। আমরা তো প্লাগিন ইনস্টল দিয়েই খালাস. . . একবার ভেবেও দেখিনা এটি আসলে কিভাবে, কখন লোড হবে। সাইটের উপর কতটা ইফেক্ট ফেলবে।

আমি নিচে একটি ডায়াগ্রাম দিলাম আশা করি আর কিছু বলতে হবেনা আমাকে।

Wordpress URL Called



Caution
This Book Not For sale

ওয়ার্ডপ্রেস প্লাগিন ডেভেলপ করার ৭টি কারন

আমি ইন্টারনেট এ “[How to Write a Plugin \(http://goo.gl/rCUY\)](http://goo.gl/rCUY)” দিয়ে অনেক সার্চ করেছি। অনেক আর্টিকেল আছে কিন্তু মন মত কয়েকটি ছাড়া বলা মত পাইনি। ওয়ার্ডপ্রেসের “কোডেক্স” আসলেই ওয়ার্ডপ্রেস সম্পর্কে জানার জন্য যথেষ্ট, এটা পরে বুঝেছি।

৭ টি কারন কেন আপনি ওয়ার্ডপ্রেস প্লাগিন বানাবেনঃ

- ১) আপনি একটি প্লাগিন সম্পর্কে জানেন কিন্তু এর বাস্তবায়ন পছন্দ করছেন না। আমরা বেশির ভাগ সময়ই এই সমস্যায় পড়ি, ২- ৩ দিন ধরে একটা প্লাগিন খোঁজার জন্য ব্যয় করি কিন্তু মন মত প্লাগিন খুঁজে পাই না। কিন্তু এই সময় এ একটা প্লাগিন লিখা হয়ে যেত।
- ২) আপনি আগের কোন প্লাগিন মডিফাই করবেন। অনেক সময়ই আপনার পছন্দ মত প্লাগিন পেলেন কিন্তু সামান্য কিছু চেঞ্জ করলেই আপনার কাজে লাগবে এক্ষেত্রে আপনাকে সেই প্লাগিনটি মডিফাই করতে জানতে হবে। অনেক সময় প্লাগিনের লেখক ব্যস্ত থাকতে পারেন এবং আপডেট করতে পারেন না। এক্ষেত্রে আপনি সেই প্লাগিনটি Author Name চেঞ্জ করে নিতে পারেন তবে [GPL Compatible license \(http://goo.gl/BOzm\)](http://goo.gl/BOzm) ঠিক রেখে আপনি পুনরায় অরিজিনাল প্লাগিনের নতুন ভার্সন বের করতে পারেন।
- ৩) কোন প্লাগিনকে এক্সটেন্ড করতে চান। অনেক সময় একটা প্লাগিন ভাল কাজ করে সাইটের জন্য কিন্তু আপনি আরো কিছু ফিচার যোগ করতে পারেন। যেমন AJAX, JQuery Effects অথবা আরো কিছু হুক লাগাতে চান যেন এটি অন্য প্লাগিনের সাথে সামঞ্জস্য থাকে। তবে অবশ্যই [GPL Compatible \(http://goo.gl/BOzm\)](http://goo.gl/BOzm) এর সাথে সামঞ্জস্য হতে হবে।
- ৪) আপনি যদি আপনার থিমের বহনযোগ্য কোড করতে চান। এটি মূলত কাস্টম থিম কোডের কথা বলা হয়েছে কোন এক্সিস্টিং থিমের।
- ৫) আপনি যদি থিম ডিজাইনার হন।
- ৬) আপনি যদি টাকা উপার্জন করতে চান প্লাগিন তৈরী করে।
- ৭) আপনি যদি আপনার সাইটের ইনকামিং লিঙ্ক বিল্ড করতে চান।

প্লাগিনের ফোল্ডার স্ট্রাকচার কেমন হবে

সকল ওয়ার্ডপ্রেস প্লাগিন wp-content/plugins ডিরেক্টরিতে ইনস্টল হয়। কিছু প্লাগিনে শুধু একটি PHP ফাইল থাকে, কিন্তু আমি সব সময় একটি ফোল্ডার করে প্লাগিন তৈরী করার পরামর্শ দেব।

প্লাগিনের স্ট্রাকচার হতে হবে এরকমঃ

Caution
This Book Not For sale

- Plugin Folder Name(আপনার প্লাগিন ফোল্ডার নাম, কোন স্পেস বা স্পেশাল ক্যারেক্টার থাকা চলবে না)
 - Main .php file
 - js ফোল্ডার(জাভাস্ক্রিপ্ট বা JQuery file এর জন্য)
 - css ফোল্ডার(স্টাইলশিট ফাইলের জন্য)
 - php/include ফোল্ডার (অন্য php ফাইল)

একটি উদাহরণ দেই ফাইলের স্ট্রাকচার কেমন হবে-

- litons-first-plugin
 - index.php
 - js
 - css
 - includes
 - images
 - uninstall.php

প্লাগিন হেডার

প্লাগিনের হেডার ছাড়া সেই ফাইল/ ফোল্ডারটিকে ওয়ার্ডপ্রেস চিনতে পারবে না। প্লাগিন হেডারই হচ্ছে একমাত্র প্রয়োজনীয় ফাংশন যা ওয়ার্ডপ্রেস দাবী করে। মানে আপনাকে প্লাগিন লেখার শুরুতে PHP comment ব্লকের মধ্যে কিছু ইনফরমেশন দিতে হবে যা একেবারে ফাইলের প্রথমে অবস্থান করে।

এখানে কিছু কথা বলে রাখি, আপনি যদি সাইটের এসইও টার্গেট করে প্লাগিন বানান তাহলে কোন ক্রমেই আপনার নাম দিয়ে প্লাগিন বানাবেন না যেমনঃ Liton's Plugin এতে এসইও তে খারাপ প্রভাব পড়ে।

নিচে উদাহরণ দেয়া হলঃ

```
<?php
```

```
/*
```

```
Plugin Name: Liton's Plugin
```

```
Plugin URI: http://liton-online.com/plugin
```

```
Description: This is a demo description.
```

```
Version: 1.0
```

```
Author: Md. Liton Arefin
```

```
Author URI: http://liton-online.com
```

```
*/
```

```
?>
```



প্লাগিনের নাম



প্লাগিনের URI



প্লাগিনের বিবরণ



প্লাগিনের ভার্সন



প্লাগিন লেখক



লেখকের URI

Caution
This Book Not For sale

<input type="checkbox"/> Plugin	Description
<input type="checkbox"/> Akismet Activate Edit Delete	Used by millions, Akismet is quite possibly the best way in the world to protect your site from comment and trackback spam. It keeps your site protected while you sleep. To get started: 1) Click the "Activate" link to the left of this description, 2) Go to your Akismet configuration page to get an API key, and 3) Go to your Akismet configuration page to activate it. Version 2.5.3 By Automattic Visit plugin site
There is a new version of Akismet available. View version 2.5.6 details or update	
<input type="checkbox"/> Liton's Plugin Activate Delete	This is a demo description. Version 1.0 By Md. Liton Arefin Visit plugin site

উপরের ছবিতে দেখুন ওয়ার্ডপ্রেসের ড্যাশবোর্ডে আমি প্লাগিন হেডারে যা যা তথ্য দিয়েছিলাম তাই দেখাচ্ছে। এটাই আপনার প্রাথমিক কাজ।

আপনি লাইসেন্সও যোগ করতে পারেন এভাবে-

```
< ?php
```

```
/* Copyright YEAR PLUGIN_AUTHOR_NAME (email : PLUGIN_AUTHOR_EMAIL)
```

```
This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.
```

```
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
```

```
*/ ? >
```

প্লাগিন পাথ

একটি কমন কাজ হল আপনার প্লাগিনের ফাইল এবং ফোল্ডারগুলোকে ওয়ার্ডপ্রেস ইনস্টলেশন ডিরেক্টরিতে রেফার করা। আপনি এই কাজটি দুই ভাবে করতে পারেন- লোকাল সার্ভার পাথ অথবা স্ট্যান্ডার্ড URL.

লোকাল সার্ভার পাথঃ

লোকাল সার্ভার পাথ আপনার কম্পিউটারের ডিরেক্টরির মতই। লোকাল সার্ভার পাথ সাধারণত ব্যবহার করা হয় যখন আপনাকে নিজের লোকাল সার্ভারের কিছু ফাইল include করতে হয়। একটি কমন প্রশ্ন হল- প্লাগিন ফাইলের পাথ দেখানোর সঠিক পদ্ধতি কোনটি? আপনার প্লাগিনের লোকাল পাথ দেখানোর জন্য আপনাকে plugin_dir_path() ফাংশন ব্যবহার করতে হবে।

plugin_dir_path() ফাংশন প্লাগিন ডিরেক্টরির ফিজিক্যাল পাথ এক্সট্রাক্ট করে ফাইলের নাম থেকে।

```
<?php plugin_dir_path($file); ?>
```

Parameters:

\$file-(string)(required)- প্লাগিন ফাইলের নাম

এখন দেখি কিভাবে প্লাগিন ফোল্ডারের নাম বের করতে হয়-

```
<?php
echo plugin_dir_path(__FILE__);
?>
```

এখানে পিএইচপি কনস্ট্যান্ট __FILE__ কে plugin_dir_path ফাংশনের মধ্য দিয়ে পাস করলাম। এই ফাংশন সার্ভার পাথ দেখাবে প্লাগিন ডিরেক্টরির।

/public_html/wp-content/plugins/liton-plugin/

একইভাবে আপনি যদি plugin_dir_path() এই ফাংশনকে যদি কোন সাব-ডিরেক্টরির ফাইলকে রেফার করি তাহলে নিচের মত করে রেফার করতে হবে।

```
<?php
echo plugin_dir_path(__FILE__).'js/scripts.js';
?>
```

এই কোডটি নিচের মত পাথ দেখাবে-

/public_html/wp-content/plugins/liton-plugin/js/scripts.js

URL পাথঃ

আরো কিছু ফাংশন আছে যেগুলো দিয়ে ওয়ার্ডপ্রেসের URL দেখানো যায়।

- plugins_url() - প্লাগিন ডিরেক্টরির পুরো URL
(যেমনঃ <http://liton-online.com/wp-content/plugins>)
- includes_url() - পুরো ডিরেক্টরির URL include করে
(যেমন <http://liton-online.com/wp-includes>)

- `content_url()` - পুরো কন্টেন্ট ডিরেক্টরি URL
(যেমনঃ <http://liton-online.com/wp-content/>)
- `admin_url()` - পুরো এডমিন URL

(যেমনঃ <http://liton-online.com/wp-admin/>)
- `site_url()` - কারেন্ট সাইট এর পুরো URL
(যেমনঃ <http://liton-online.com/>)
- `home_url()` - কারেন্ট সাইট এর পুরো হোম URL
(যেমনঃ <http://liton-online.com/>)

পাথ নিয়ে আরো বিস্তারিত দেখতে পারেনঃ

http://codex.wordpress.org/Determining_Plugin_and_Content_Directories
http://codex.wordpress.org/Function_Reference/plugins_url
http://codex.wordpress.org/Function_Reference/plugin_dir_url
http://codex.wordpress.org/Function_Reference/plugin_dir_path

`plugins_url()` ফাংশন আপনার উত্তম বন্ধু হতে পারে যখন আপনি প্লাগিন ডেভেলপ করবেন।

```
<?php
plugins_url($path, $plugin);
?>
```

Parameters:

`$path`-(string)(optional)- রিলেটিভ পাথ URL
`$plugin`- (string)(optional)- প্লাগিন ফাইল যা আপনি (`__FILE__`) এর মাধ্যমে
 পাস করতে চান।

যেমনঃ

```
<?php
echo '';
?>
```

এর পুরো URL হবে

```

```

অধ্যায়- তিনঃ হুকস

ওয়ার্ডপ্রেস হুক(Hook) কি?

ওয়ার্ডপ্রেসে হুক বলতে সাধারণত `add_action()` অথবা `add_filter()` ফাংশন কে বোঝায়। এই দুটি ফাংশনকে `remove_action()` অথবা `remove_filter()` দিয়ে মুছে ফেলা যায়। যদিও অনেকের কাছে শুনেছি যে এই দুটি ফাংশনের মধ্যে কোন পার্থক্য নেই। কিন্তু আসলে তা নয়, বিরাট পার্থক্য আছে এ দুটি ফাংশনের মধ্যে। আমি মূলত এই দুটি ফাংশন দিয়েই কিছু প্লাগিন বানিয়ে দেখাব। কারন খুবই গুরুত্বপূর্ণ এই দুটি ফাংশন সম্পর্কে ভালোভাবে জানা।

ফিল্টার হুক (Filter Hooks)

ফিল্টার হুক ফাংশনের কাজ হচ্ছে কনটেন্টকে মডিফাই করা। ফিল্টার হুক দিয়ে টাইটেল, কনটেন্ট ইত্যাদি মডিফাই করা যায়। যেমন টাইটেল এ কোন ট্যাগ না থাকা, ভিজিটর কমেণ্টে `<> tag, span tag` ইত্যাদি দিয়ে সাইটের লেআউটের পরিবর্তন করে দিতে পারে মানে কোন কমেণ্টে যদি এরকম `ht ml` কোড পার্স করা থাকে তাহলে দেখতে খারাপ লাগবে। এগুলো অবশ্য এখন ওয়ার্ডপ্রেস অটোমেটিক্যালি করে। কিন্তু এই কাজগুলো মূলত ফিল্টার হুক দিয়ে করা হয়ে থাকে।

ফিল্টার হুক দুই ধরনের `add_filter()` এবং `remove_filter()` ফাংশন।

খুব গুরুত্বপূর্ণ একটি কথা ফিল্টার হুককে রিটার্ন করতে হয়।

```
<?php
add_filter($tag, $function, $priority, $accepted_args);
?>
```

- `$tag`- যে হুককে আপনি রেজিস্টার করতে চান।
- `$function`- যে ফাংশনকে আপনি আউটপুটে ফিল্টার করে দেখাতে চান।
- `$priority`- একটি ইন্টিজার সংখ্যা যা আপনার ফিল্টারকে ক্রমানুসারে সাজাবে। যদি কোন সংখ্যা না দেয়া থাকে তাহলে এটি ১০ ডিফল্ট হিসেবে থাকে।
- `$accepted_args`- যে সংখ্যক প্যারামিটার একটি ফাংশন গ্রহণ করতে পারে। ডিফল্ট হিসেবে ১ থাকে। আপনার ফাংশনে অন্ততঃ একটি প্যারামিটার থাকতে হবে যা রিটার্ন করবে।

```
apply_filter($tag,$value)
```

- `$tag`- ফিল্টার হুকের নাম
- `$value`- হুকের মধ্য দিয়ে যে প্যারামিটার পাস করাবেন।

আমরা এখন একটি উদাহরণ দেখব যেখানে টাইটেল এর প্রতিটা ওয়ার্ডকে বড় হাতের অক্ষরে রূপান্তর করব। নিচে একটি প্লাগিন দেখুন-

```
<?php
/*
```

```
Plugin Name: Liton filter
```

```
Plugin URI: http://liton-online.com/plugins
```

```
Description: This is a demo description.
```

```
Version: 1.0
```

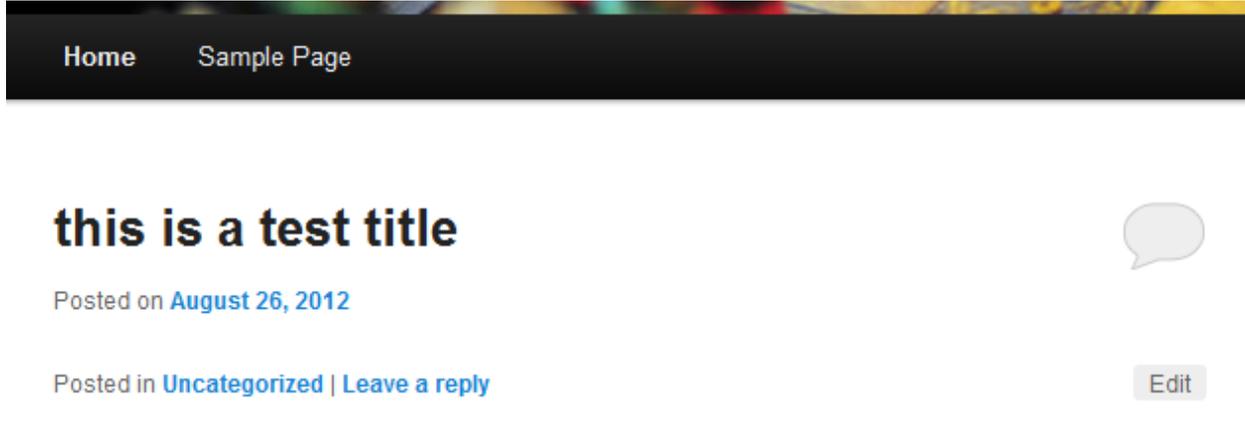
Caution
This Book Not For sale

```

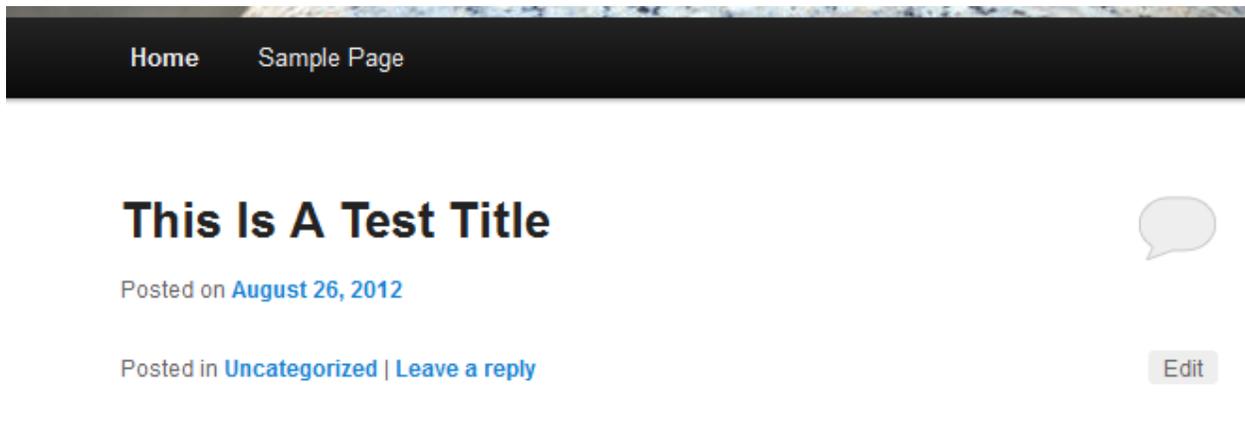
Author: Md. Liton Arefin
Author URI: http://liton-online.com
*/
add_filter('the_title','liton_modify_title');
function liton_modify_title($content)
{
return ucwords($content);
}
?>

```

প্লাগিনটি এক্টিভেট করার আগে পোস্ট টাইটেল দেখুন-



এবার প্লাগিনটি এক্টিভেট করার পরে পোস্ট টাইটেল দেখুন-



এই প্লাগিনটি আমাদের অনেক সময়ই কাজে লাগতে পারে। কারণ অনেকেই আমরা এই ফরম্যাটটি পছন্দ করি। হঠাৎ করেই আমরা পোস্ট টাইটেল লিখার সময় বড় হাতের অক্ষর করতে ভুলে যেতে পারি। আমরা এই কাজটি খুব ছোট একটি প্লাগিন দিয়ে অটোমেটিকভাবে করতে পারি।

Caution
This Book Not For sale

এখন আমি কিছু অতিরিক্ত বিষয় নিয়ে আলোচনা করব, যেগুলো মেনে চললে ভবিষ্যতের জন্য ভাল হবে।

আমি একটি ফাংশন লিখেছি-

```
add_filter('the_title','liron_modify_title');
function liron_modify_title($content)
{
return ucwords($content);
}
```

একটু বিস্তারিত বললে, আমরা the_title এর মধ্য দিয়ে যা পাস করব তা ফিল্টার করবে।

এখানে আমরা ucwords কে the_title এর মধ্য দিয়ে পাস করেছি। এবার আরো শর্ট ফর্ম দেখি।

```
add_filter('the_title', ucwords);
```

তার মানে পুরো প্লাগিনটি হবে এরকম-

```
<?php
/*
```

Plugin Name: Liron filter

Plugin URI: <http://liron-online.com/plugins>

Description: This is a demo description.

Version: 1.0 Author: Md. Liron Arefin

Author URI: <http://liron-online.com>

```
*/
```

```
add_filter('the_title', ucwords);
```

```
?>
```

কি খুব সহজ লাগছে? এখানে বলে রাখি যে, ucwords এটি রিসিভ করবে the_title যা পাস করছে এবং অটোমেটিক্যালি রিটার্ন করবে the_title এ। সুতরাং এখানে রিটার্ন এর দরকার নেই।

এবার আপনি যদি পিএইচপি ভার্সন ৫.৩ ব্যবহার করেন তাহলে নিচের মত করে anonymous function ব্যবহার করবেন। আপনারা জাভাস্ক্রিপ্ট, জেকুয়েরি'র সাথে পরিচিত। এটি অনেকটাই এরকম।

```
<?php
/*
```

Plugin Name: Liron filter

Plugin URI: <http://liron-online.com/plugins>

Description: This is a demo description.

Version: 1.0 Author: Md. Liron Arefin

Author URI: <http://liron-online.com>

```
*/
```

```
add_filter('the_title', function($content) {
```

```
return ucwords($content);
```

```
});
```

```
?>
```

ভাল করে দেখুন তো পার্থক্যটা কোথায়? নিজে নিজে কাজ করলে এভাবে করবেন অবশ্যই। এটি একটু এডভান্স লেবেলের কাজ। যদিও প্লাগিন বানানোর ক্ষেত্রে এটি এখনো আদর্শ নয় কারণ অনেকেরই পিএইচপি ভার্সন এখনো ৪ এই থেকে গেছে। সুতরাং এই পদ্ধতিটি একান্তই নিজের জন্য।

এ্যাকশন হুক (Action Hooks)

Action Hooks হচ্ছে সে সমস্ত হুক যেগুলো ওয়ার্ডপ্রেসের লাইফ সাইকেল নিয়ে কাজ করে। Action Hooks দুই ধরনের `add_action()`, `remove_action()`। এই হুকগুলো কাজ করে যখন কোন পোস্ট সেভ করা হয়, পোস্ট পাবলিশ করা হয়, কমেট করা হয় কোন ব্লগ পোস্টে অথবা ওয়েবসাইটে হেডার/ফুটার জেনারেট হয়। এগুলো প্রতিটি একটি একটি করে Actions, যখন ইউজার তার এডমিনিস্ট্রেটর মেনুতে প্রবেশ করে এগুলো সবই Action Hooks। এখানে আমরা কনটেন্টকে ফিল্টার করছি, আমরা ওয়ার্ডপ্রেসের নির্দিষ্ট ফাংশনগুলোকে হুক করে কাজ করছি।

আমরা একটি উদাহরণ দেখি-

```
<?php
/*
Plugin Name: Liton Action
Plugin URI: http://liton-online.com/plugins
Description: This is a demo description.
Version: 1.0 Author: Md. Liton Arefin
Author URI: http://liton-online.com
*/
add_action('wp_footer', function($content) {
echo "Hello, This is Liton Arefin";
});
?>
```

আপনাদের সুবিধার্থে বোঝার জন্য নিচের মত করে লিখুন-

```
<?php
/*
Plugin Name: Liton Action
Plugin URI: http://liton-online.com/plugins
Description: This is a demo description.
Version: 1.0 Author: Md. Liton Arefin
Author URI: http://liton-online.com
*/
add_action('wp_footer', 'liton_footer_action');
function liton_footer_action($content) {
echo "Hello, This is Liton Arefin";
}
?>
```

আমরা খুব ছোট একটি উদাহরণ দিয়ে দেখলাম যে, action হুক কিভাবে কাজ করে।
নিচে দেখি সাইটে কেমন দেখায়-

Proudly powered by WordPress

Hello, This is Liton Arefin

উপরের ছবিতে দেখুন, আমি Hello, This is Liton Arefin এই লাইনটি প্রিন্ট করেছি এবং তা ফুটারের বাম পাশে একেবারে নিচে দেখাচ্ছে। এবার সোর্স কোড এ দেখি আসলেই এর পজিশনটা কোথায় আছে...

```
<div id="site-generator">
    <a href="http://wordpress.org/" title="Semantic Personal Publishing Platform" rel="gen
</div>
</footer><!-- #colophon -->
</div><!-- #page -->

Hello, This is Liton Arefin<script type='text/javascript' src='http://localhost/dev-liton/wp-includes/js/admin-bar.js?'
```

উপরে দেখুন, ফুটারের ঠিক নিচেই লাইনটি দেখাচ্ছে।

এবার আরেকটি বাস্তব এবং খুবই গুরুত্বপূর্ণ উদাহরণ দেখি। এই প্লাগিনটির কাজ হবে যখনই কোন কমেন্ট পোস্ট হবে তখনই এডমিনকে মেইল পাঠাবে যে নতুন কমেন্ট পোস্ট হয়েছে। যদিও ওয়ার্ডপ্রেস এই কাজটি করে থাকে। আমরা এই কাজটি ম্যানুয়ালী করে দেখি।

```
<?php
/*
```

Plugin Name: Liton Comments Mail

Plugin URI: http://liton-online.com/plugins

Description: This is a demo description.

Version: 1.0 Author: Md. Liton Arefin

Author URI: http://liton-online.com

```
*/
```

```
add_action('comment_post', function() {
$email= get_bloginfo('admin_email');
wp_mail(
    $email,
    'New Comment Posted!!!',
    'A new comment has been left on your blog.'
);
});
```

```
?>
```

এখানে comment_post নামের হুকে একটি ভেরিয়েবল \$email –এ এডমিন মেইলকে স্টোর করেছি যাতে করে ডাইনামিক ভাবে সে যেকোন সাইটের এডমিন মেইলকে ডিটেস্ট করতে পারে। তবে এক্ষেত্রে যদি get_bloginfo('admin_email'); ব্যবহার করা হত তাহলে এডমিন মেইলকে ডিটেস্ট করত কিন্তু wp_mail(email,subject,message); এর মত করে লিখেছি ভাল করে দেখুন। প্লাগিনটি এক্টিভেট করে দেখুন। আপনার সাইট যদি অনলাইনে থাকে তাহলে নিজেই একটি কমেন্ট করে দেখুন, আপনার এডমিন মেইলে একটি মেইল চলে গেছে।

অধ্যায়- চারঃ প্লাগিন কোডিং স্ট্যান্ডার্ড

প্লাগিন এক্টিভেশন/ডিএক্টিভেশন ফাংশন

আপনি যখন একটি প্লাগিনকে এক্টিভ করেন তখনই এক্টিভেশন ফাংশন ট্রিগার হয়। এই ফাংশনকে `register_activation_hook()` বলে।

```
<?php register_activation_hook($file, $function); ?>
```

Parameters:

- \$file- প্রাইমারি প্লাগিন ফাইলের পথ
- \$function- যে ফাংশনটি ট্রিগার হবে যখন প্লাগিন এক্টিভেট হয়।

যেমনঃ-

```
<?php
register_activation_hook(__FILE__, 'liton_plugin_install');
function liton_plugin_install()
{
//do something cool stuff
}
?>
```

আমরা একটি প্লাগিন বানাই এভাবে-

```
<?php
/*
Plugin Name: Liton Version
Plugin URI: http://liton-online.com/plugins
Description: This is a demo description.
Version: 1.0 Author: Md. Liton Arefin
Author URI: http://liton-online.com
*/
register_activation_hook(__FILE__, 'liton_plugin_install');
function liton_plugin_install()
{
If(version_compare(get_bloginfo('version'),'5.1','<'))
{
deactivate_plugins(basename(__FILE__));
//deactivate our plugin
}
}
?>
```

যখন আমরা প্লাগিন ডিএক্টিভেট করি তখন প্লাগিন ডিএক্টিভেশন ফাংশন কল হয়।

```
<?php
register_deactivation_hook ($file, $function); ?>
```

প্লাগিনের কোডগুলো এরকম হবে-

```
<?php
register_deactivation_hook (__FILE__, 'liron_plugin_install');
function liron_plugin_install()
{
//do something
}
?>
```

আরেকটি গুরুত্বপূর্ণ বিষয় হল আনইনস্টল। ওয়ার্ডপ্রেস আপনাকে প্লাগিন আনইনস্টল এর জন্য দুটি অপশন দিচ্ছে- uninstall.php file and the uninstall hook.

uninstall.php

এটি মোটামুটি গ্রহনযোগ্য পদ্ধতি। এ মেথড ব্যবহার করার জন্য আপনাকে uninstall.php নামে একটি ফাইল তৈরি করতে হবে আপনার প্লাগিন ডিরেক্টরিতে। ফাইলটি নিচের মত হবে-

```
<?php
// যদি ওয়ার্ডপ্রেস এক্সিট করার সাথে আনইনস্টল কল করা না হয়
If(!defined('WP_UNINSTALL_PLUGIN'))
exit();
delete_option('liron_plugin_options');
// অতিরিক্ত অপশন এবং কাস্টম টেবিল মুছে ফেলুন
?>
```

Uninstall Hook

দ্বিতীয় পদ্ধতি হল Uninstall Hook। যদি আপনি কোন প্লাগিনকে ডিলিট করতে চান এবং uninstall.php ফাইল না থাকে তাহলে এই হুক এক্সিকিউট হয়(যদি থাকে)।

```
<?php
register_uninstall_hook($file,$function);
?>
```

Caution
This Book Not For sale

কোডিং স্ট্যান্ডার্ড

প্লাগিন বানাবেন কিন্তু আপনার কোড কেমন হবে? চিন্তা আসতে পারে। ওয়ার্ডপ্রেস আপনার জন্য একটি স্ট্যান্ডার্ড দিয়ে দিয়েছে যা আপনাকে ওয়ার্ডপ্রেসের জন্য প্লাগিন লিখতে গেলে ফলো করতেই হবে।

http://codex.wordpress.org/WordPress_Coding_Standards

বেসিক একটি উদাহরণ দিলাম নিচে-

```
<?php
/**
 * Short description
 *
 * Longer more detailed description
 *
 * @param type $varname1 Description
 * @param type $varname2 Description
 * @return type Description
 */
function boj_super_function( $varname1, $varname2 ) {
//do function stuff
}
?>
```

পিএইচপি স্ট্যান্ডার্ড

আমরা অনেকেই পিএইচপির সর্ট ফর্ম(<? ?>) ব্যবহার করি। এটা সম্পূর্ণরূপে বর্জন করুন। কারণ অনেক হোস্টিং কোম্পানিতেই এটা ডিফল্ট হিসেবে বন্ধ করে দেয়া থাকে। স্ট্যান্ডার্ড ট্যাগ ব্যবহার করুন(<?php ?>) .

এসকিউএল স্টেটমেন্ট

প্লাগিন তৈরির সময় আপনাকে অনেক সময়ই ডাটাবেস থেকে ইনফরমেশন নিতে হতে পারে। অনেক লাইনের কোড লিখতে হতে পারে সেক্ষেত্রে ভাল প্র্যাকটিস হচ্ছে এসকিউএল এর কমান্ড গুলো বড় হাতের অক্ষরে লেখা। যেমন-

```
SELECT username FROM table1 WHERE status = 'active'
```

Caution
This Book Not For sale

অধ্যায়- পাঁচঃ ওয়ার্ডপ্রেসের সাথে ইন্টিগ্রেশন

মেনু এবং সাব মেনু তৈরি করণ

আমরা অনেক প্লাগিনেই দেখি যে, একটি মেনু আইটেম হিসেবে আমাদের ড্যাশবোর্ডে দেখতে পাই এবং সেই মেনুতে ক্লিক করলে সেটিংস পেজ আসে সেখানে প্লাগিনের যাবতীয় সেটিংস দিয়ে আমরা প্লাগিন কনফিগার করি। এবার এই চমৎকার কাজটি করব আমরা।

টপ লেভেল মেনু

প্রথম পদক্ষেপ হচ্ছে একটি টপ লেভেল মেনু তৈরি করা যেটি আমরা ড্যাশবোর্ডে মেনু লিস্টে একটি নতুন আইটেম হিসেবে যোগ করব। মেনু তৈরি করার জন্য আপনাকে `add_menu_page()` ফাংশন ব্যবহার করতে হবে।

```
<?php
```

```
add_menu_page( page_title, menu_title, capability, menu_slug, function, icon_url, position );
```

```
?>
```

`add_menu_page()` ফাংশন নিচের প্যারামিটারগুলো সাপোর্ট করে

- `page_title`- আপনার পেজের টাইটেল কি হবে `<title>` tag এর মধ্যে।
- `menu_title`- ড্যাশবোর্ডে আপনার মেনুর নাম।
- `capability`- সর্বনিম্ন `capability` লাগবে একটি মেনু ভিউ হতে।
- `menu_slug`- স্লাগ নাম যা মেনুকে রেফার করে- তবে এটি ইউনিক হতে হবে।
- `function`- একটি পেজের কনটেন্ট দেখতে যে ফাংশন কল করা হয়।
- `icon_url`- একটি কাস্টম ইমেজ লিঙ্ক মেনু আইকন দেখানোর জন্য।
- `position`- মেনুর পজিশন অর্ডার যেখানে এটা দেখানো হবে।

এবার আমরা একটি বাস্তব উদাহরণ দেখব-

```
<?php
```

```
/*
```

```
Plugin Name: Liton Menu
```

```
Plugin URI: http://liton-online.com/plugins
```

```
Description: This is a demo description.
```

```
Version: 1.0 Author: Md. Liton Arefin
```

```
Author URI: http://liton-online.com
```

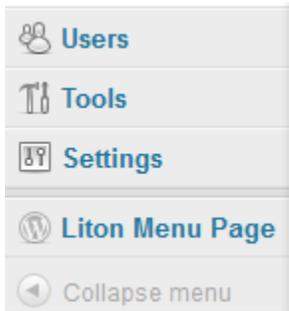
```
*/
```

```
add_action( 'admin_menu', 'liton_menu' );
```

```
function liton_menu() {
```

```
//create custom top-level menu
add_menu_page( 'Liton Menu Page', 'Liton Menu Page', 'manage_options',
__FILE__, 'liton_menu_settings_page', plugins_url( '/images/wp-logo.png',
__FILE__ ) );
}
?>
```

আমরা একটি টপ লেভেল মেনু তৈরি করলাম। দেখুন ড্যাশবোর্ডে কেমন দেখায়-



Liton Menu Page নামে একটি মেনু তৈরি হয়ে গেছে। এবারে আমরা সাবমেনু করব। মানে Liton Menu Page এর নিচে কিছু সেটিংস মেনু(যদি প্রয়োজন হয়)।

সাব মেনু রেজিস্টার করার জন্য `add_submenu_page()` ফাংশন ব্যবহার করা হয়।

```
<?php
```

```
add_submenu_page( parent_slug, page_title, menu_title, capability, menu_slug,
function );
```

```
?>
```

- `parent_slug`- মেনু স্লাগ যা আমরা আগে ডিফাইন করেছি।
- `menu_slug`- যা মেনুকে সাবমেনুতে রেফার করবে।

বাকি ফাংশনগুলো আগের মতই।

এবার সাবমেনুর জন্য কিছু অতিরিক্ত কোড যোগ করি।

```
//create submenu items
```

```
add_submenu_page( __FILE__, 'About My Plugin', 'About', 'manage_options',
__FILE__.'_about', liton_menu_about_page );
add_submenu_page( __FILE__, 'Help with My Plugin', 'Help',
'manage_options',
__FILE__.'_help', liton_menu_help_page );
add_submenu_page( __FILE__, 'Uninstall My Plugin', 'Uninstall',
'manage_options',
__FILE__.'_uninstall', liton_menu_uninstall_page );
```

পুরো কোডটি হবে এরকম-

```
<?php
/*
```

```
Plugin Name: Liton Menu
```

```
Plugin URI: http://liton-online.com/plugins
```

```
Description: This is a demo description.
```

```
Version: 1.0 Author: Md. Liton Arefin
```

```
Author URI: http://liton-online.com
```

```
*/
```

```
add_action( 'admin_menu', 'liton_menu'
);
```

```
function liton_menu() {
```

```
//create custom top-level menu
```

```
add_menu_page( 'Liton Menu Settings Page', 'Liton Menu Page',
'manage_options', __FILE__,
```

```
'liton_menu_settings_page',
```

```
plugins_url( '/images/wp-logo.png', __FILE__ ) );
```

```
//create submenu items
```

```
add_submenu_page( __FILE__, 'About My Plugin', 'About', 'manage_options',
__FILE__.'_about', liton_menu_about_page );
```

```
add_submenu_page( __FILE__, 'Help with My Plugin', 'Help',
'manage_options',
```

```
__FILE__.'_help', liton_menu_help_page );
```

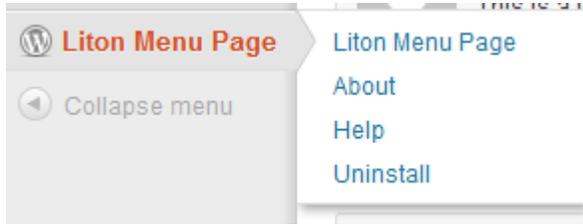
```
add_submenu_page( __FILE__, 'Uninstall My Plugin', 'Uninstall',
'manage_options',
```

```
__FILE__.'_uninstall', liton_menu_uninstall_page );
```

```
}
```

```
?>
```

নিচে দেখুন সাবমেনুগুলো তৈরি হয়ে গেছে-



মেনুকে আগের মেনুর সাথে যুক্ত করা

এবার আসি দ্বিতীয় পদ্ধতিতে, যদি আপনার প্লাগিনের একটি মাত্র পেজ থাকে। একটি মাত্র পেজেই আপনার সকল সেটিংস হয়ে যায় তাহলে এই পদ্ধতি গ্রহন করুন। যেমন, শেয়ারিং সাইটগুলোর জন্য শুধু আপনার আইডি/ফ্যানপেজের আইডি দরকার হয় এরকম প্লাগিন। এধরনের প্লাগিনের জন্য আপনার অতিরিক্ত মেনু তৈরি করার দরকার নেই বরং এতে করে ইউজার বিরক্ত বোধ করতে পারেন। এ ধরনের প্লাগিনগুলো সেটিংস এর সাবমেনু হিসেবে থাকে।

add_options_page() এই ফাংশন দিয়ে এসব মেনু রেজিস্টার করা হয়।

```
<?php
add_options_page( page_title, menu_title, capability,
menu_slug, function);
?>
```

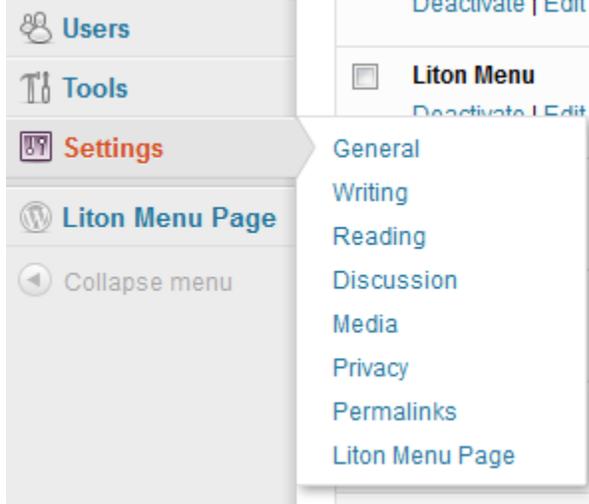
এবার চলুন এরকম একটি মেনু তৈরি করি-

```
<?php
/*
Plugin Name: Liton Sub Menu
Plugin URI: http://liton-online.com/plugins
Description: This is a demo description.
Version: 1.0 Author: Md. Liton Arefin
Author URI: http://liton-online.com
*/
```

```
add_action( 'admin_menu', 'liton_submenu');
function liton_submenu() {
//create a submenu under Settings
add_options_page( 'Liton Menu Settings Page', 'Liton Menu Page',
'manage_options', __FILE__,
'liton_menu_settings_page');
}
```

?>

সাবমেনু তৈরি হয়ে গেছে, চলুন দেখি ড্যাশবোর্ডে কেমন দেখায়-



একেবারে নিচের দিকে দেখুন Liton Menu Page নামে একটি মেনু তৈরি হয়েছে।

উইজেট তৈরি করা

উইজেট তৈরি করার জন্য ওয়ার্ডপ্রেস WP-Widget ক্লাস দিয়ে তৈরি করা হয়। উইজেট এর একটি ফরম্যাট দেখুন নিচে।

```
<?php
class My_Widget extends WP_Widget {
function My_Widget() {
// processes the widget
}
function form($instance) {
// displays the widget form in the admin dashboard
}
function update($new_instance, $old_instance) {
// process widget options to save
}
function widget($args, $instance) {
// displays the widget
}
}
?>
```

উইজেট রেজিস্টার করার জন্য widget_init() ফাংশনকে হুক করা হয়। চলুন একটি উইজেট তৈরি করি-

```
<?php
/*
Plugin Name: Liton Widget
Plugin URI: http://liton-online.com/plugins
Description: This is a demo description.
Version: 1.0 Author: Md. Liton Arefin
Author URI: http://liton-online.com
*/
class Messenger extends WP_Widget
{
function __construct()
{
$params=array(
        'description'      =>'Display message to readers',
        'name'              => 'Message'
    );
    parent::__construct('Messenger', ' ', $params);
}

public function form()
{

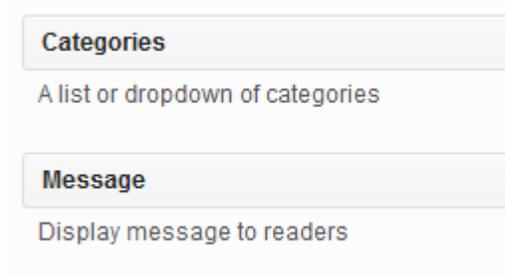
}

public function widget()
{

}

}
add_action('widgets_init','liton_register_message');
function liton_register_message()
{
register_widget('Messenger');
}
?>
```

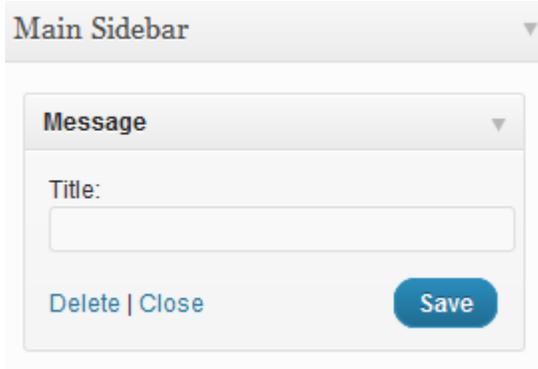
খুব সাধারণ একটি উইজেট প্লাগিন, এটি। আমরা ইউজারদেরকে একটি মেসেজ দেখাবো সাইডবারে। সেটাই করার চেষ্টা করছি। এখন প্লাগিনটি এক্টিভেট করে দেখি-



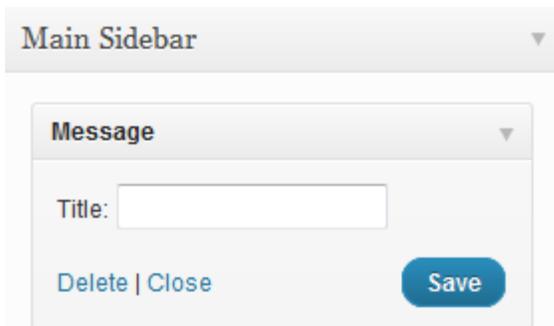
আমাদের উইজেটটি দেখাচ্ছে। এবার আরো কিছু ফাংশনাল কাজ করি। এটি যদি সাইডবারে ড্রাগ করে দেন তাহলে কোন কাজ করবে না। কারন আমরা এখনো কিছুই করিনি। ফর্ম ফাংশনে কিছু কাজ করি যা উইজেট টাইটেল হিসেবে দেখাবে-

```
public function form($instance)
{
    extract($instance);
    ?>
    <p>
    <label for="<?php echo $this->get_field_id('title');?>">Title: </label>
    <input
        class="widefat"
        id="<?php echo $this->get_field_id('title'); ?>"
        name="<?php echo $this->get_field_name('title'); ?>"
        value="<?php if(isset($title)) echo esc_attr($title); ?>" />
    </p>
    <?php
}
}
```

আমরা উপরে class="widefat" ব্যবহার করেছি। এই ক্লাস ব্যবহার করার কারনে ফর্ম এর ইনপুট ফিল্ডটি অনেক সুন্দর হয়েছে।



যদি এই ক্লাস ব্যবহার না করি তাহলে কেমন দেখাবে নিচে দেখুন-



আমাদের উইজেট প্লাগিনটির কোড নিচে দেওয়া হল

```
<?php
/*
```

```
Plugin Name: Liton Widget
```

```
Plugin URI: http://liton-online.com/plugins
```

```
Description: This is a demo description.
```

```
Version: 1.0 Author: Md. Liton Arefin
```

```
Author URI: http://liton-online.com
```

```
*/
```

```
class Messenger extends WP_Widget
```

```
{
```

```
function __construct()
```

```
{
```

```
$params=array(
```

```
    'description' => 'Display message to readers',
```

```
    'name' => 'Message'
```

```
);
```

```
parent::__construct('Messenger','',$params);
```

```

}

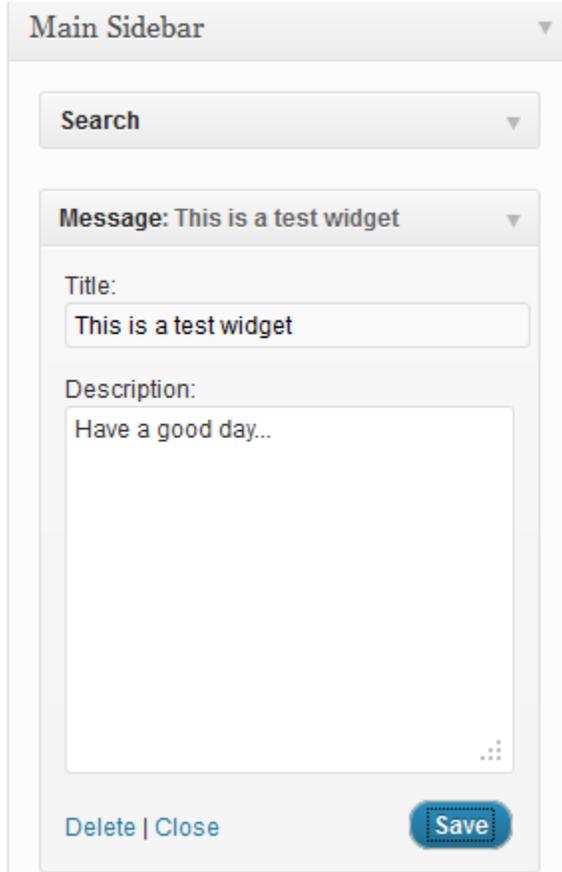
public function form($instance)
{
extract($instance);
//print_r($instance);
?>
<p>
<label for="<?php echo $this->get_field_id('title');?>">Title: </label>
<input
    class="widefat"
    id="<?php echo $this->get_field_id('title'); ?>"
    name="<?php echo $this->get_field_name('title'); ?>"
    value="<?php if(isset($title)) echo esc_attr($title); ?>" />
</p>

<p>
<label for="<?php echo $this->get_field_id('description');?>">Description:
</label>
<textarea
    class="widefat"
    rows="10"
    id="<?php echo $this->get_field_id('description'); ?>"
    name="<?php echo $this->get_field_name('description'); ?>"><?php
if(isset($description)) echo esc_attr($description); ?></textarea>
</p>
<?php
}
public function widget($args,$instance)
{
extract($args);
extract($instance);
echo $before_widget;
echo $before_title.$title.$after_title;
echo "<p>$description</p>";
echo $after_widget;
}
}
add_action('widgets_init','liton_register_message');
function liton_register_message()

```

```
{
register_widget('Messenger');
}
?>
```

এবার আমরা ড্যাশবোর্ড থেকে একটি মেসেজ দেব ভিজিটরদেরকে-



আমরা এই উইজেটটি ডাইনামিক ভাবে করেছি কারন এখানে `<h2>`, `<h3>` `<h4>` এ ধরনের কোন ট্যাগ আমরা ব্যবহার করিনি। আমরা থিম ডেভেলপ করছিনা সুতরাং এ কাজটি আমাদের অন্যভাবে করা দরকার ছিল, এটি থিম ডেভেলপার দেখিয়ে দেবেন যে কেমন ট্যাগ হবে সাইডবারে। সাইটের সাইডবারে কেমন দেখায় তা দেখুন-
আমরা দুটি থিমে এটি টেস্ট করে দেখব, প্রথমটি টুয়েন্টি ইলেভেন

Search

THIS IS A TEST WIDGET

Have a good day..

AUGUST 2012

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

« Jun

এবার আরেকটি থিমে দেখি-

 Search

THIS IS A TEST WIDGET

Have a good day...

AUGUST 2012

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19

অধ্যায়- ছয়ঃ প্লাগিন ফাংশন

কিছু বেসিক ফাংশন

__() ফাংশন

__() ফাংশনটি পিএইচপিতে আপনার টেক্সটকে রেডি করে ট্রান্সলেশন এবং রিটার্ন করে। উদাহরণ হিসেবে, আপনি একটি ভ্যালুকে __() ফাংশনের মধ্যে ভেরিয়েবল দিয়ে assign করুন। মনে রাখবেন এই ফাংশনটি দুটি আন্ডারস্কোর() দিয়ে গঠিত, একটি নয়।

```
<?php
$text=__('Wordpress is a wonderful publishing platform','liron_plugin');
?>
```

_e() ফাংশন

_e() ফাংশনটি আপনার টেক্সটে লোকালাইজ করে। এটা পিএইচপিতে echo এর মত কাজ করে, কোন টেক্সটকে স্ক্রিনে দেখানোর জন্য।

এবার একটু মজা করি যা হুকে দেখেছি আগেই-

_e() ফাংশনটি দিয়ে আগের প্লাগিনটি এডিট করব,

```
<?php
/*
Plugin Name: Liron Action
Plugin URI: http://liron-online.com/plugins
Description: This is a demo description.
Version: 1.0 Author: Md. Liron Arefin
Author URI: http://liron-online.com
*/
add_action('wp_footer', 'liron_footer_action');
function liron_footer_action() {
/* Output the translated text. */
_e('This site runs off the coolest platform ever &mdash;
WordPress.',
'liron-plugin');
}
?>
```

এবার আউটপুট দেখি

Proudly powered by WordPress

This site runs off the coolest platform ever — WordPress.

esc_attr__() ফাংশন

এই ফাংশনটি esc_attr() এর মত। এটি এইচ.টি.এম.এল. এট্রিবিউটকে escape করে, আমরা এবার esc_attr__() এবং __() এই দুটি ফাংশনকে একটি করে প্লাগিন তৈরি করব।

```
<?php
/*
```

Plugin Name: Liton Esc Attr

Plugin URI: <http://liton-online.com/plugins>

Description: This is a demo description.

Version: 1.0 Author: Md. Liton Arefin

Author URI: <http://liton-online.com>

```
*/
```

```
/* A function that returns a link to the site's terms of service
page. */
```

```
function liton_terms_of_service_link() {
return '<a href="http://liton-online.com/tos" title="" .
esc_attr__( 'Visit the Terms of Service page', 'boj-plugin' ) ."'>' .__( 'Terms of
Service', 'liton-plugin' ) . '</a>';
}
```

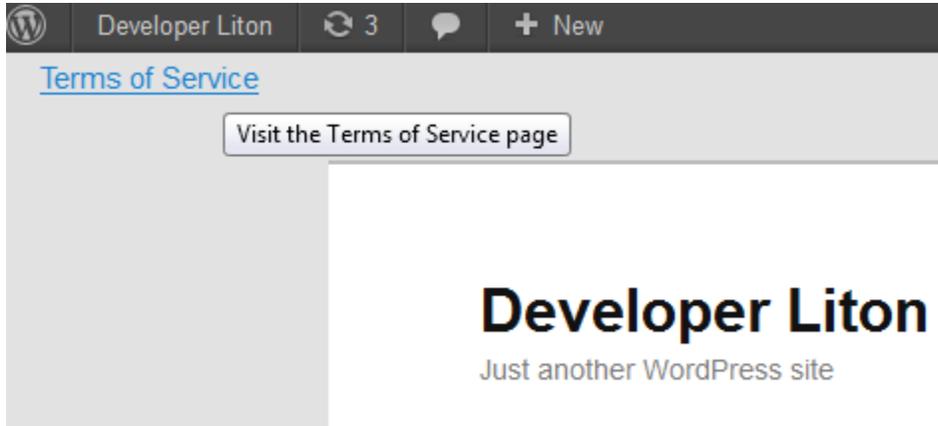
```
/* Display the output of the liton_terms_of_service_link()
```

```
function. */
```

```
echo liton_terms_of_service_link();
```

```
?>
```

প্লাগিনটির আউটপুট দেখুন-



এরকম আরো কিছু ফাংশন আছে যেগুলো আপনারা দেখে নেবেন ইন্টারনেট ঘেটে।

- `esc_html__()` Function
- `esc_attr()` Function
- `esc_url()` Function
- `esc_js()` Function
- `esc_html()` Function
- `esc_attr_e()` Function
- `esc_html_e()` Function
- `_x()` Function
- `_ex()` Function
- `esc_attr_x()` Function
- `esc_html_x()` Function
- `_n()` Function
- `_nx()` Function
- `_n_noop()` Function
- `_nx_noop()` Function

Translation এর জন্য কিছু টুলসঃ

- Poedit: <http://poedit.net>
- GlotPress: <http://glotpress.org>
- Launchpad: <https://translations.launchpad.net>
- Pootle: <http://pootle.locamotion.org>
- KBabel: <http://i18n.kde.org>
- GNU Gettext: <http://gnu.org/software/gettext>

GlotPress এখন ওয়েব বেসড টুলস হিসেবে ব্যবহার করা হচ্ছে ওয়ার্ডপ্রস প্রজেক্টের পেছনে একজন অথবা টিমের জন্য ট্রান্সলেটিং সফটওয়্যার হিসেবে।

অধ্যায়- সাতঃ প্লাগিন সিকিউরিটি

সিকিউরিটি

কম্পিউটারের ভাষায় “সিকিউরিটি” মানে ভীতিকর একটি শব্দ যেমন Cross Site Scripting(XSS), Cross Site Request Forgery(CSRF), SQL Injection, Privilege, Escalation, Vulnerabilities এবং Holes.

দুর্বল কোড একটি সার্ভারের সুরক্ষার জন্য হুমকীর কারন হতে পারে।

প্লাগিনের সিকিউরিটি কি?

আপনার প্লাগিনটি যেন সিকিউর হয় vulnerabilities, data integrity এবং reliability থেকে।

আপনার প্লাগিনকে কি সিকিউরিটি দেবেন?

ওয়ার্ডপ্রেস এনভায়রনমেন্টে আপনার প্লাগিনকে সিকিউরিটি দেয়া খুব কঠিন কাজ নয় এবং খুব সময় সাপেক্ষ বিষয়ও নয়। ওয়ার্ডপ্রেসের নিজস্ব কিছু ফাংশন এক্ষেত্রে আপনাকে খুবই সাহায্য করবে।

ইউজার পারমিশনস

আমরা অনেক প্লাগিন ইনস্টল করার পরেই দেখেছি যে যখনই এডমিন প্যানেল এ প্রবেশ করতে যাই তখন একটি এরর মেসেজ আসে। মেসেজটি নিচের মত-

You do not have sufficient permissions to manage plugins for this site.

লেখা আছে- You do not have sufficient permissions..... এটা নিশ্চয়তা দেয় যে যেকোন একশন সীমাবদ্ধ থাকে নির্দিষ্ট কিছু রাইটস দিয়ে। ওয়ার্ডপ্রেসের একটি কমন ফাংশন হল `current_user_can()`. একটি উদাহরণ দেই-

```
<?php
// Capability:
if ( !current_user_can('install_plugins') )
wp_die( 'Insufficient permissions' );
// Role:
if( !current_user_can('editor') )
wp_die( 'You cannot edit this setting' );
?>
```

খুব তাড়াতাড়ি আপনি এই `current_user_can()` ফাংশন চেক করবেন না কারণ এটা নির্ভর করবে `get_currentuserinfo()` এই ফাংশনের উপরে, আর এর ব্যবহারে সীমাবদ্ধতা আছে। এটা একটি প্লাগেবল ফাংশন(pluggable function) আর এর লোকেশন হচ্ছে `wp-includes/pluggable.php` যা প্লাগিন এক্টিভ হবার পর লোড হয়।

Nonces

কেউ এটিকে আবার Nonsense(নির্বোধ) ভাববেন না,এটি ওয়ার্ডপ্রেসের ভাষায় Nonces । কম্পিউটারের ভাষায় nonce বা cryptographic nonce হচ্ছে “number used once. ” এটা ছোট এবং স্পষ্ট কিছু র্যানডম স্ট্রিং, যেমন- পাসওয়ার্ড, যা কিছু নির্দেশ করে। **Nonce** হিডেন ফিল্ড ফর্মকে Retrieve বা display করে-

- একজন ওয়ার্ডপ্রেস ইউজার
- যেকোন একটি একশন (delete, update, save, and such)
- একটি অবজেক্ট(একটি পোস্ট, প্লাগিন সেটিংস ইত্যাদি)
- ২৪ ঘন্টার একটি সময় ফ্রেম।

যেমন, পোস্ট #50 ডিলিট করবেন আপনার ওয়ার্ডপ্রেস ব্লগ থেকে তাহলে আপনার লিঙ্কটি হবে এরকম-

http://example.com/wp-admin/post.php?post=43&action=trash&_wpnonce=83a08fcbc2.

এখানে 83a08fcbc2 এটি হল nonce যা ২৪ ঘন্টা স্থায়ী থাকে। শুধুমাত্র এটি যদি আপনার দ্বারা ব্যবহৃত হয় এবং ডিলিট হয় পোস্ট #50, যখন আপনি লিঙ্কে ক্লিক করেন।

সবচেয়ে গুরুত্বপূর্ণ হল যে Nonce কোন malicious ইউজার বুঝতে পারেনা এবং সঠিক লিঙ্ক ছাড়া nonce এর পেজে চলে যায় যেখানে পারমিশন থাকে না।

আরো বিস্তারিত দেখুন

http://codex.wordpress.org/WordPress_Nonces

Nonce URL তৈরি

এবার চলুন আগের উদাহরণ থেকে Nonce URL তৈরি করি একটি পোস্ট ডিলিট করার জন্য wp_nonce_url() ফাংশন ব্যবহার করা হয়।

```
<?php
$url = wp_nonce_url( $url, $action );
?>
```

প্রথম প্যারামিটার \$url হচ্ছে স্ট্রিং URL এড্রেস এর যাকে আপনি কুয়েরী স্ট্রিং এর Nonce করতে চান। অব্যবহৃত প্লাগিনের ডিলিট লিঙ্ক এরকম থাকে-

```
http://example.com/wp-admin/edit.php?page=liton
_utags&liton_action=delete&id=6;
```

এখানে দেখুন প্যারামিটার liton_action এবং এর ট্যাগ আইডি।

দ্বিতীয় প্যারামিটার হচ্ছে \$action একটি স্ট্রিং যা আপনি একটি অবজেক্ট এর জন্য একটি action তৈরি করবেন। উপরে দেখলাম delete এ্যাকশন এর ট্যাগ আইডি 6, সুতরাং \$action প্যারামিটার হতে পারে 'liton_utags-delete_tag6'। নিচে বাস্তব উদাহরণ দেখি-

```
<?php
$delete_url = add_query_arg(array('liton_action'=>'delete','id'=>$id));
$nonced_url = wp_nonce_url( $delete_url, 'liton_utags-delete_tag'.$id );
?>
```

```
<a href="<?php echo $nonced_url; ?>">delete</a>
```

এই ট্যাগটি হবে ডিলিট লিঙ্ক। এখানে add_query_arg() ফাংশন ব্যবহার করা হয়েছে যা কারেন্ট URL কে এ্যারে প্যারামিটার এর মধ্যে রাখে। অন্যভাবে বললে এটি যোগ করে এরকমঃ

```
? liton_action=delete&id=6 অথবা & liton_action=delete&id=6
```

Nonce Form তৈরি

Nonce ফর্মকেও প্রোটেক্ট করে wp_nonce_field() ফাংশনের মাধ্যমে।

```
<?php
wp_nonce_field( $action, $name, $referer, $echo ); ?>
```

- \$action (string) (optional) - action নাম। ইউনিক identifier। optional কিন্তু recommended।
- \$name (string) (optional) – হিডেন ফিল্ড ফর্ম ভেরিয়েবল নাম।
- \$referer (boolean) (optional) – ডিফল্ট হিসেবে true থাকে। যেখানে ফিল্ড ভেলিডেশন এর দরকার হয়।

- \$echo (boolean) (optional) - ডিফল্ট হিসেবে true থাকে। যেখানে আউটপুট দেখানোর বা হিডেন ফর্ম ফিল্ড রিটার্ন করার জন্য দরকার হয়।

ফর্মে আপনার action এর নাম দিন।

```
<form method="post">
  <!-- some inputs here ... -->
  <?php wp_nonce_field('name_of_my_action','name_of_nonce_field'); ?>
</form>
```

আমরা একটি প্লাগিন বানাবো যেখানে অব্যবহৃত ট্যাগগুলোকে আমরা ম্যানুয়ালি ডিলিট বা রিনেম করতে পারব। এটি একটু বড় কিন্তু অনেক বিষয়ই আগেই আলোচনা করা হয়েছে। দেখলেই বুঝতে পারবেন।

একটি বাস্তবধর্মী উদাহরন নিচে দিলাম-

```
<?php
/*
Plugin Name: Liton Unused Tags
Plugin URI: http://liton-online.com/plugins
Description: This is a demo description.
Version: 1.0 Author: Md. Liton Arefin
Author URI: http://liton-online.com
*/
// Add an entry for our option page to the Posts menu
add_action('admin_menu', 'liton_utags_add_page');

function liton_utags_add_page() {
add_posts_page( 'Unused Tags', 'Unused Tags','manage_options','liton_utags',
'liton_utags_option_page' );
}

// Catch any liton_action parameter in query string
add_action( 'admin_init', 'liton_utags_do_action' );
// Proceed to requested liton_action if applicable
function liton_utags_do_action() {
if( !isset( $_REQUEST['liton_action'] ) )
return;
if( !current_user_can( 'manage_options' ) )
wp_die( 'Insufficient privileges!' );
```

```

$tid = $_REQUEST['id'];
$action = $_REQUEST['liron_action'];

if( $action == 'done' ) {
add_action( 'admin_notices', 'liron_utags_message' );
return;
}
check_admin_referer( 'liron_utags-'.$action.'_tag'.$tid );
switch( $action ) {
case 'rename':
$newtag = array( 'name' => $_POST['name'], 'slug' =>
$_POST['name'] );
wp_update_term( $tid, 'post_tag', $newtag );
break;
case 'delete':
wp_delete_term( $tid, 'post_tag' );
break;
}

wp_redirect( add_query_arg( array( 'liron_action' => 'done' )
) );
}
// Admin notice
function liron_utags_message() {
echo "<div class='updated'><p>Action
completed</p></div>";
}
// Draw the tag management page
function liron_utags_option_page() {
?>
<div class="wrap">
<?php screen_icon(); ?>
<h2>Unused Tags</h2>

<?php
if( $tags = liron_utags_find_orphans() ):
echo '<p>You currently have '.count( $tags ). ' unused
tags:</p>';
echo '<ol>';
foreach( $tags as $tag ) {

```

```

$tid = $tag->term_id;
$name = esc_attr( $tag->name );
$delete_url= add_query_arg(
array('liron_action'=>'delete','id'=>$tid) );
$nonced_url= wp_nonce_url( $delete_url,
'liron_utags-delete_tag'.$tid );
?>
<li>
<form action="" method="post">

<?php wp_nonce_field( 'liron_utags-rename_tag'.$tid );
?>
<input type="hidden" name="liron_action"
value="rename" />
<input type="hidden" name="id" value="<?php echo
$tid; ?>" />
<input type="text" name="name" value="<?php echo
$name; ?>" />
<input type="submit" value="Rename" /> or
<a href="<?php echo $nonced_url; ?>">delete</a>
this tag
</form>
</li>
<?php }
else: ?>
<p>You have no unused tags.</p>

<?php endif; ?>
</ol>
</div>
<?php
}
// Find unused tags, return them in an array
function liron_utags_find_orphans() {
global $wpdb;
$sql = "SELECT terms.term_id, terms.name FROM
{$wpdb->terms} terms
INNER JOIN {$wpdb->term_taxonomy} taxo
ON terms.term_id=taxo.term_id
WHERE taxo.taxonomy = 'post_tag'

```

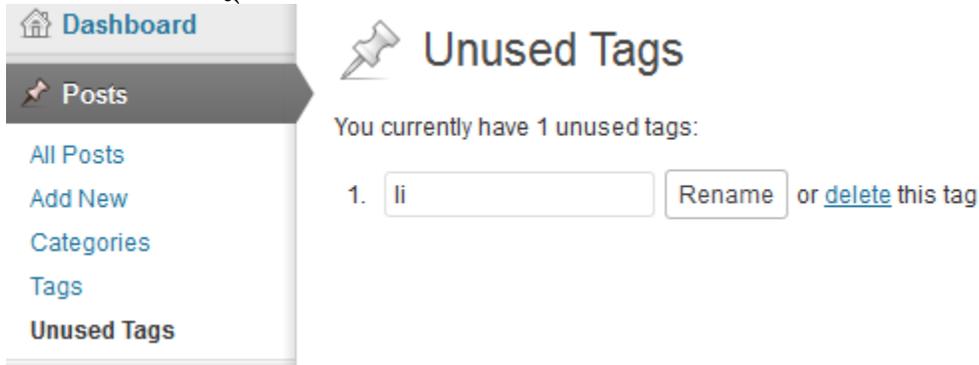
```
AND taxo.count=0";
```

```
return $wpdb->get_results( $sql );
}
?>
```

ড্যাশবোর্ডে এই প্লাগিনের মেনু পাওয়া যাবে পোস্ট এর সাবমেনু হিসেবে। নিচে দেখুন-



এবার প্লাগিন মেনুতে ক্লিক করে দেখি প্লাগিনের ভেতরের কাজ-



উপরে দেখুন আমি একটি ট্যাগ তৈরি করেছিলাম। i নামে। এটি এখানে দেখাচ্ছে। আমি চাইলে এটি এডিট বা ডিলিট করতে পারি।

এবার ডিলিট লিঙ্ক এ মাউসের কার্সর রেখে নিচে লিঙ্ক দেখুন বা ডিলিট করুন একটি ট্যাগ।

আমার এখানে URL টি এরকম

http://localhost/dev-liton/wp-admin/edit.php?page=liton_utags&liton_action=done&id=4&wpnonce=30a44b7986

এখানে আমি liton_action=done ব্যবহার করেছি।

Nonces এর ফাংশনগুলোর নাম নিচে দিলাম। আপনারা দেখে নিবেন ইন্টারনেট থেকে।

```
wp_nonce_field()
wp_nonce_url()
wp_verify_nonce()
wp_create_nonce()
```

```
check_admin_referer()
check_ajax_referer()
```

ডাটাবেস

আপনারা যেহেতু পিএইচপি নিয়ে কাজ করেন, এসকিউএল ডাটাবেস এর সাথে সবাই কমবেশী পরিচিত। এর সিনট্যাক্স গুলো মোটামুটি সবার জানা। একই ভাবে অনেকেই ওয়ার্ডপ্রেসেও ডাটাবেস কোড লিখেন, এটা কোডিং এর জন্য খারাপ দিক। আমি নরমাল পিএইচপি- এসকিউএল কোড এর উদাহরন দিচ্ছি-

```
<?php
$sql="INSERT INTO $wpdb->posts(post_title) VALUES('$title')";
$wpdb->query($sql); ?>
```

এবার ওয়ার্ডপ্রেসের স্ট্যান্ডার্ড কোডিং সিস্টেম দেখুন-

```
<?php
$wpdb->insert($wpdb->posts,
              array('post_title' => $title),
              array('%s')
);
?>
```

ডাটাবেস সিকিউর করার জন্য আপডেট অপারেশন দেখাব-

নিচের কোডটি খারাপ প্র্যাকটিস

```
<?php
$wpdb->query("UPDATE $wpdb->posts SET post_title='$newTitle' WHERE
ID=$my_id");
?>
```

উপরের মত করে না লিখে নিচের মত করে লিখুন-

```
<?php
$wpdb->update($wpdb->posts,
              Array('post_title' => '$newTitle'),
              Array('ID' => $my_id
);
?>
```

অনুরূপ ভাবে অন্যান্য অপারেশন গুলো লিখবেন।

অধ্যায়- আটঃ লোকালাইজেশন এবং Settings API

লোকালাইজেশন

লোকালাইজেশন একটি গুরুত্বপূর্ণ বিষয়। এর ফাংশনগুলো হল-

load_theme_textdomain()

load_plugin_textdomain()

উদাহরণ-

```
<?php $var=__('Hello','domain'); ?>
```

```
<?php _e('Hello','domain'); ?>
```

বিস্তারিত নিচের লিঙ্কে দেখুন-

http://codex.wordpress.org/I18n_for_WordPress_Developers

সেটিংস API

সেটিংস API হচ্ছে আপনি কিছু অপশন তৈরি করবেন যা ইন্টারনাল ভাবে আপনার প্লাগিনে এপ্লাই হবে।

যখন আপনি ইউজার- ডিফাইন অপশন তৈরি করবেন আপনার প্লাগিনের জন্য তখন Settings API –এ আপনার কোড যতটা পারুন সহজ করুন।

Settings API আপনার যে কাজগুলো handle করে-

- ওয়ার্ডপ্রেসকে জানিয়ে দেয় যে আপনি নিজস্ব অপশন ব্যবহার করবেন
- আপনার ইনপুটকে ফাংশনের মধ্যে উল্লেখ করবে
- বেশির ভাগ অপশন পেজ নিজেই তৈরি করবে
- \$_POST ডাটা গুলোকে মনিটর করে
- অপশন তৈরি এবং আপডেট করে(যদি প্রয়োজন হয়)

Settings API ফাংশন

Settings API ফাংশনে তিনটি স্টেপ থাকে-

১। প্রথমে ওয়ার্ডপ্রেসকে বলে যে আপনি নতুন সেটিংস ব্যবহার করবেন আপনার প্লাগিনের জন্য।

২। এর পরে সেটিংস ডিফাইন করুন (text areas, input boxes, and any HTML form element) এবং এদেরকে একসাথে গ্রুপ করে রাখে।

৩। ওয়ার্ডপ্রেসকে আপনার সেটিংস actual form এ দেখানোর জন্য বলে।

এবার আমরা একটি প্লাগিন বানাবো Administration Page –এ। রেফারেন্স হিসেবে দেখতে পারেন-

http://codex.wordpress.org/Adding_Administration_Menus

http://codex.wordpress.org/Function_Reference/add_options_page

http://codex.wordpress.org/Action_Reference#Actions_Run_During_an_Admin_Page_Request

প্লাগিনের এডমিন পেজ হবে এরকম-

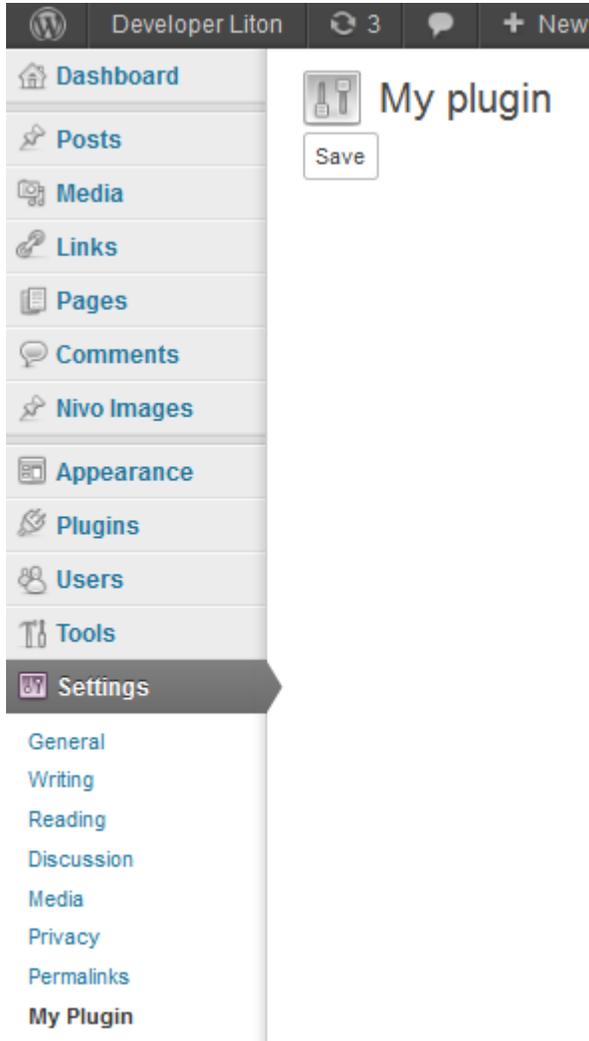
http://example.com/wp-admin/options-general.php?page=liton_myplugin

প্রথমে আমরা মেনু তৈরি করব আগের মত করে-

```
// Add a menu for our option page
add_action('admin_menu', 'liton_myplugin_add_page');
function liton_myplugin_add_page() {
    add_options_page( 'My Plugin', 'My Plugin', 'manage_options', 'liton_myplugin',
    'liton_myplugin_option_page');
}
```

```
// এবার অপশন পেজ এ ফর্ম নিয়ে কিছু সেটিংস দেব
function liton_myplugin_option_page() {
    ?>
    <div class="wrap">
    <?php screen_icon(); ?>
    <h2>My plugin</h2>
    <form action="options.php" method="post">
    <?php settings_fields('liton_myplugin_options'); ?>
    <?php do_settings_sections('liton_myplugin'); ?>
    <input name="Submit" type="submit" value="Save" />
    </form></div>
    <?php
    }
    ?>
```

উপরের কোডে আমরা একটি ফর্মের মধ্যে শুধু সাবমিট বাটন এবং My Plugin নামে পেজ তৈরি করেছি। ড্যাশবোর্ডে দেখুন-



সেটিংস এর সাব মেনু হিসেবে My Plugin আছে এবং My Plugin তে ক্লিক করলে আমরা উপরের পেজ তৈরি করেছি। যেখানে সাবমিট বাটনটি দেখাচ্ছে। এবার তিনটি প্যারামিটার রেজিস্টার করব-

```
<?php
register_setting(' liton _myplugin_options',' liton _myplugin_options',' liton
_myplugin_validate_options');
?>
```

প্রথমটি সেটিংস গ্রুপের নাম, দ্বিতীয়টি অপশন নাম যা আমরা get_option() ফাংশন দিয়েও পেতে পারি। তৃতীয়টি কলব্যাক ফাংশন। এটি একটি স্ট্রিং রেফারেন্স ফাংশন, এখানে আমরা নাম

দিয়েছি `liton_myplugin_validate_options` যা পাস করবে ফর্মের সকল সেভ সেটিংস। আমরা এটা পরে ডিফাইন করব।

সেটিংস ডিফাইন সেকশন

`add_settings_field()` এবং `add_settings_section()` ফাংশন দিয়ে আমরা আমাদের সকল সেটিংস ডিফাইন করব।

```
<?php
add_settings_section( 'liton_myplugin_main', 'My Plugin Settings',
'liton_myplugin_section_text', 'liton_myplugin');
add_settings_field( 'liton_myplugin_text_string', 'Enter text here',
'liton_myplugin_setting_input', 'liton_myplugin', 'liton_myplugin_main');
?>
```

`add_settings_section()` ফাংশন দিয়ে আমরা দেখাব আসলে পেজ এ সেকশন কেমন দেখাবে। এর চারটি প্যারামিটার আছে-

- সেকশনের জন্য HTML ID ট্যাগ
- সেকশন টাইটেল `<H3>` ট্যাগের মধ্যে
- কলব্যাক ফাংশন এর নাম
- সেটিংস পেজ এর যে সেকশনটি দেখাবে (`?page=liton_myplugin` এটি URL এর অংশ হিসেবে থাকবে)

দ্বিতীয় প্যারামিটার `add_settings_field()` ফর্মের ইনপুট হিসেবে দেখানো হয়েছে। এখানে পাঁচটি প্যারামিটার আছে-

- সেকশনের জন্য HTML ID ট্যাগ
- পরের ফিল্ডে কি প্রিন্ট করা হবে তা টেক্সট হিসেবে দেখানো হয়েছে('Enter text here')
- কলব্যাক ফাংশন ফর্মে প্রিন্ট করবে
- সেটিংস পেজ এর যে সেকশনটি দেখাবে
- সেটিংস পেজের সেকশন ফিল্ড যা `add_settings_section()` ফাংশনে ডিফাইন করা হয়েছে।

```
<?php
// Explanations about this section
function liton_myplugin_section_text() {
echo '<p>Enter your settings here.</p>';
```

```

}
// Display and fill the form field
function liton_myplugin_setting_input() {
// get option 'text_string' value from the database
$options = get_option( ' liton _myplugin_options' );
$text_string = $options['text_string'];
// echo the field
echo "<input id='text_string'
name=' liton _myplugin_options[text_string]'
type='text' value='{ $options['text_string']}' />";
}
?>

```

দ্বিতীয় ফাংশন অপশন ভ্যালু text_string কে এগারেতে স্টোর করে রেখেছে।
এবার ইউজার ইনপুট ভ্যালিডেট করব-

```

<?php
function liton_myplugin_validate_options( $input ) {
$valid = array();
$valid['text_string'] = preg_replace('/[^a-zA-Z]/',"",$input['text_string'] );
return $valid;
}
?>

```

ফর্মকে রেন্ডার করব-

```

<?php
// Draw the options page
function liton_myplugin_options_page() {
?>
<div class="wrap">
<?php
screen_icon();
?>
<h2>My plugin</h2>
<form action="options.php" method="post">
<?php
settings_fields(' liton_myplugin_options');
do_settings_sections(liton_myplugin);

```

```

?>
<input name="Submit" type="submit" value="Save " />
</form></div>
<?php
} ?>
settings_fields() ফাংশনটি রেফারেন্স হিসেবে ব্যবহার করা হয়েছে যা আমরা
register_setting() ফাংশনে ডিক্লেয়ার করেছি।
do_settings_sections() ফাংশনটি কল করবে সকল আউটপুট সেকশন এবং ফর্ম ফিল্ডকে যা
আমরা পূর্বে ডিফাইন করেছি।
আমাদের কাজ শেষ !!!
এবার পুরো প্লাগিনটি দেখব-
<?php
/*
Plugin Name: Settings API example
Plugin URI: http://liton-online.com/plugins
Description: This is a demo description.
Version: 1.0 Author: Md. Liton Arefin
Author URI: http://liton-online.com
*/
// Add a menu for our option page
add_action('admin_menu', 'liton_myplugin_add_page');
function liton_myplugin_add_page() {
add_options_page( 'My Plugin', 'My Plugin', 'manage_options', 'liton_myplugin',
'liton_myplugin_option_page');
}
// Draw the option page
function liton_myplugin_option_page() {
?>
<div class="wrap">
<?php screen_icon(); ?>
<h2>My plugin</h2>
<form action="options.php" method="post">
<?php settings_fields('liton_myplugin_options'); ?>
<?php do_settings_sections('liton_myplugin'); ?>
<input name="Submit" type="submit" value="Save
Changes" />
</form></div>
<?php
}

```

```

// Register and define the settings
add_action('admin_init', 'liton_myplugin_admin_init');
function liton_myplugin_admin_init()
{
register_setting( 'liton_myplugin_options',
'liton_myplugin_options',
'liton_myplugin_validate_options' );
add_settings_section( 'liton_myplugin_main', 'My Plugin
Settings',
'liton_myplugin_section_text', 'liton_myplugin' );
add_settings_field( 'liton_myplugin_text_string', 'Enter text
here',
'liton_myplugin_setting_input', 'liton_myplugin',
'liton_myplugin_main' );
}
// Draw the section header
function liton_myplugin_section_text() {
echo '<p>Enter your settings here.</p>';
}
// Display and fill the form field
function liton_myplugin_setting_input() {
// get option 'text_string' value from the database

$options = get_option( 'liton_myplugin_options' );
$text_string = $options['text_string'];
// echo the field
echo "<input id='text_string'
name='liton_myplugin_options[text_string]'
type='text' value='$text_string' />";
}
// Validate user input (we want text only)
function liton_myplugin_validate_options( $input ) {
$valid = array();
$valid['text_string'] = preg_replace(
'/[^a-zA-Z]/',
'',
$input['text_string'] );
return $valid;
}
?>

```

দেখুন ড্যাশবোর্ডে কেমন দেখায়-

The screenshot shows a WordPress dashboard interface for a plugin named 'My plugin'. At the top, there is a yellow notification box that says 'Settings saved.'. Below this, the heading 'My Plugin Settings' is displayed. Underneath the heading, there is a prompt 'Enter your settings here.' followed by a text input field with the value 'HELLO'. To the left of the input field is a label 'Enter text here'. Below the input field is a 'Save' button.

হয়ে গেল আমাদের প্রথম পূর্ণাঙ্গ প্লাগিন অপশন পেজ সহ। আরেকটি জিনিস যেটার জন্য এতকিছু, প্লাগিন URL এড্রেস-

http://localhost/dev-liton/wp-admin/options-general.php?page=liton_myplugin&settings-updated=true

আমার এখানে এরকম দেখাচ্ছে টেক্সট বক্সে কোন কিছু লিখে সেভ করার পর।

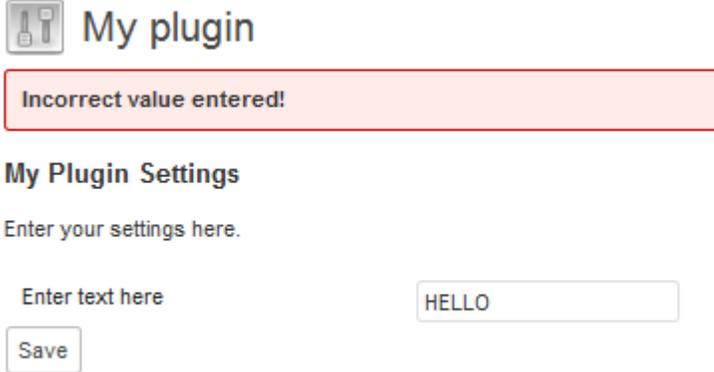
এবারে আরেকটি জিনিস দেখব, আমরা অনেক সময়ই অনেক প্লাগিনের ইনপুট ফিল্ডে ভুল ডাটা ইনপুট করলে এরর মেসেজ দেখায়।

এই কাজটি করব এবার। আমরা এখানে `esc_attr()` বা `esc_html()` ব্যবহার করিনি সুতরাং আমাদের লিমিটেশন থাকছে। ভেলিডেট ফাংশনটি একটু এডিট করি নিচের মত করে।

```
// Validate user input (we want text only)
function liton_myplugin_validate_options( $input ) {
    $valid['text_string'] = preg_replace(
        '/[^a-zA-Z]/',
        '',
        $input['text_string'] );
    if( $valid['text_string'] != $input['text_string'] ) {
        add_settings_error('liton_myplugin_text_string','liton_myplugin_texterror',
```

```
'Incorrect value entered!','error');
}
return $valid;
}
```

উপরে দেখুন `add_settings_error()` ফাংশন ব্যবহার করেছি যা এরর চেক করবে ফর্মের। আমি একটি ভুল ডাটা(স্পেশাল ক্যারেক্টার) ইনপুট করে দেখলাম এরর দেখাচ্ছে-



My plugin

Incorrect value entered!

My Plugin Settings

Enter your settings here.

Enter text here

তার মানে আমরা সফল হয়েছি। আপনার প্লাগিনটি এডিট করে নেবেন। কোর সেকশন এবং ফিল্ডের লিস্ট দিলাম আপনাদের সুবিধার জন্য।

WordPress Settings Pages	Section Names	Field Set Names
General Settings (options-general.php)	'general'	'default'
Writing Settings (options-writing.php)	'writing'	'default'
		'remote_publishing'
		'post_via_email'
Reading Settings (options-reading.php)	'reading'	'default'
Discussion Settings (options-discussion.php)	'discussion'	'default'
		'avatars'
Media Settings (options-media.php)	'media'	'default'
		'embeds'
		'uploads'

WordPress Settings Pages	Section Names	Field Set Names
Privacy Settings (options-privacy.php)	'privacy'	'default'
Permalink Settings (options-permalink.php)	'permalink'	'optional'

CSS/JS ফাইল যুক্ত করা

আপনার প্লাগিনে সিএসএস বা জাভাস্ক্রিপ্ট যোগ করতে হতে পারে। আপনি চাইলে বাহিরের কোন বা আপনার তৈরি কোন সিএসএস বা জাভাস্ক্রিপ্ট ফাইল যোগ করতে পারেন। তবে এক্ষেত্রে আমরা যেভাবে HTML এ ফাইল যোগ করি সেরকম না।

```
function liton_scripts()
{
echo '<script type="text/javascript" src="path/to/jquery.js"></script>';
echo '<script type="text/javascript" src="path/to/script.js"></script>';
}
add_action('wp_head','liton_scripts');
```

এটি সম্পূর্ণরূপে ভুল পদ্ধতি। নিচে দেখুন ওয়ার্ডপ্রেসে কিভাবে ফাইল যুক্ত করতে হয়-

```
function liton_scripts()
{
wp_enqueue_script('jquery');
wp_enqueue_script('theme-scripts','path/to/scripts.js');
}
add_action('wp_enqueue_script','liton_scripts');
```

অনুরূপ ভাবে আপনি স্টাইলশীট যোগ করতে পারেন-

```
function liton_styles()
{
wp_enqueue_style('css');
wp_enqueue_style('theme-styles','path/to/styles.css');
}
```

রেফারেন্স দেখুন-

http://codex.wordpress.org/Function_Reference/wp_enqueue_script

http://codex.wordpress.org/Function_Reference/wp_enqueue_style

সর্টকোড

সর্টকোড হচ্ছে ওয়ার্ডপ্রেসের স্পেসিফিক স্পেশাল ট্যাগ যা আপনি পোস্ট বা পেজের মধ্যে ইনসার্ট করে অন্য কোন কনটেন্টকে ওয়েবসাইটের মধ্যে রিপ্লেস করতে পারবেন। যদি আপনি গ্যালারি টাইপের কোন প্লাগিন ব্যবহার করে থাকেন তাহলে আপনার ধারণা থাকার কথা। যাই হোক আমি কয়েকটি উদাহরণ দেখাব।

প্রথমে একটি প্লাগিন তৈরি করব যেখানে add_shortcode() নামের একটি ফাংশন থাকবে। এটি দুটি প্যারামিটার গ্রহণ করে।

প্রথমটি, শর্টকোডের নাম।

দ্বিতীয়টি, যে ফাংশনকে কল করবেন সেটি। আমরা একটি উদাহরণ দেখি-

```
<?php
/*
```

```
Plugin Name: Liton Shortcode
```

```
Plugin URI: http://liton-online.com/plugins
```

```
Description: This is a demo description.
```

```
Version: 1.0 Author: Md. Liton Arefin
```

```
Author URI: http://liton-online.com
```

```
*/
```

```
add_shortcode('liton','hi');
```

```
function hi()
```

```
{
```

```
echo "Hello World !!!";
```

```
}
```

```
?>
```

উপরে দেখুন খুব ছোট কিন্তু আপনি চাইলে এর ব্যবহার অনেক বিস্তৃত করতে পারেন। এখানে সর্টকোড নাম হচ্ছে liton আর এটিকে লেখা হয় এভাবে [liton]

আমরা পোস্টের মধ্যে ইনসার্ট করে দেখি আউটপুট কেমন দেখায়-



এবার আউটপুট দেখি-

Posted on August 27, 2012

[Edit](#)

Shortcode Test

Hello World !!!

আমাদের তৈরি করা সর্টকোড সফলভাবেই কাজ করছে।
 আরেকটি উদাহরণ দেখব, আপনারা অনেকেই গুগল এডসেন্স নিয়ে কাজ করেন। আপনারা অনেকেই এডসেন্স এর জন্য প্লাগিন ব্যবহার করেন কিন্তু চাইলেই এই কাজটি আপনারা নিজের মত করে সাজিয়ে নিতে পারেন। গুগলের কাছে সব পোস্ট ভ্যালু পায়না, আপনারা যারা এক্সপার্ট আছেন তারা এটা ভাল করে জানেন। তাই অনেকেই গুগলের কোড সব পেজে দেখাতে চান না। এজন্য প্লাগিন ব্যবহার করে লাভ হয়না কারন প্লাগিনে জায়গা ফিল্ড থাকে। এতে করে আর্নিং ও কম হয়। আপনারা সর্টকোড ব্যবহার করে যেকোন জায়গায় এডস দেখাতে পারেন।
 নিচে আমি দুটি সর্টকোড তৈরি করেছি দু'ধরনের এডস দেখানোর জন্য।

```
<?php
/*
Plugin Name: Liton Adsense Shortcode
Plugin URI: http://liton-online.com/plugins
Description: This is a demo description.
Version: 1.0 Author: Md. Liton Arefin
Author URI: http://liton-online.com
*/
add_shortcode('adsense_468','adsense_468_60');
add_shortcode('adsense_250','adsense_250_250');
function adsense_468_60($atts)
{
    return '<script type="text/javascript"><!--
```

```

google_ad_client = "ca-pub-0956247412222369";
/* 468x60, created 10/15/10 */
google_ad_slot = "4372705450";
google_ad_width = 468;
google_ad_height = 60;
//-->
</script>
<script type="text/javascript"
src="http://pagead2.googlesyndication.com/pagead/show_ads.js">
</script>
';
}

function adsense_250_250($atts)
{
return '<script type="text/javascript"><!--
google_ad_client = "ca-pub-0956247412222369";
/* 250x250, created 8/18/09 */
google_ad_slot = "9368091893";
google_ad_width = 250;
google_ad_height = 250;
//-->
</script>
<script type="text/javascript"
src="http://pagead2.googlesyndication.com/pagead/show_ads.js">
</script>
';
}
?>

```

একটি সর্টকোড ব্যানার টাইপের জন্য এবং আরেকটি ২৫০*২৫০ সাইজের। এবার পোস্টে যেকোন একটি সর্টকোড ইনসার্ট করুন যেখানে আপনার দেখানোর প্রয়োজন মনে করেন। আমি দুটি কোড টেস্ট করে দেখলাম-

Shortcode Test

Permalink: <http://localhost/dev-liton/?p=>

Upload/Insert 

B *I* ABC    

[adsense_250]

[adsense_468]

এবার সাইটের আউটপুট দেখি-

Shortcode Test



দুটি এডসই দেখাচ্ছে। আরো অনেক কাজে আপনি সর্টকোড ব্যবহার করতে পারেন।
রেফারেন্সঃ

http://codex.wordpress.org/Shortcode_API

অধ্যায়- নয়ঃ প্লাগিন উদাহরন

ওয়ার্ডপ্রেসের ডেট/টাইম/কমেন্ট নাম্বার বাংলা করার প্লাগিন

আমাকে কয়েকজন বলেছিলেন ওয়ার্ডপ্রেসের ডিফল্ট ডেট/টাইম/কমেন্ট নাম্বার বাংলা হয়না, ল্যাংগুয়েজ বাংলা করার পরও ডেট/টাইম/কমেন্ট নাম্বার গুলো ইংরেজি থেকে যায়। নিচের প্লাগিনটি ব্যবহার করে দেখতে পারেন-

```
<?php
/*
Plugin Name: Liton Bangla date/time/comment number
Plugin URI: http://liton-online.com/plugins
Description: Bangla date/time/comment number
Version: 1.0 Author: Md. Liton Arefin
Author URI: http://liton-online.com
*/
/**
 * This function converts all english number's to bangla number
 */
function make_bangla_number($str)
{
    $engNumber = array(1,2,3,4,5,6,7,8,9,0);
    $bangNumber =
array( ' ১' , ' ২' , ' ৩' , ' ৪' , ' ৫' , ' ৬' , ' ৭' , ' ৮' , ' ৯' , ' ০' );
    $converted = str_replace($engNumber, $bangNumber, $str);

    return $converted;
}

add_filter( 'get_the_time', 'make_bangla_number' );
add_filter( 'the_date', 'make_bangla_number' );
add_filter( 'get_the_date', 'make_bangla_number' );
add_filter( 'comments_number', 'make_bangla_number' );
add_filter( 'get_comment_date', 'make_bangla_number' );
add_filter( 'get_comment_time', 'make_bangla_number' );
?>
```

স্লাইডার প্লাগিন

আমরা এই বইয়ের সকল টিউটোরিয়ালে যা যা শিখলাম তা এবার একটি পূর্ণাঙ্গ প্লাগিনের এপ্লাই করব। স্লাইডার প্লাগিন সম্পর্কে আমরা সবাই জানি। একটি ওয়েবসাইটের চেহারা পালটে দেয় একটি স্লাইডার। অনেক ডাইনামিক মনে হয়, দেখতেও অনেক ভাল লাগে। এই প্লাগিনে আমরা এডমিন প্যানেল থেকে ইমেজ যোগ করব এবং তা আমাদের সাইটে দেখাব। চলুন কাজ শুরু করি-
প্রথমে নিভো স্লাইডার এর ওয়েবসাইটে যান এবং [jQuery Plugin](#) ডাউনলোড করুন

```
<?php
```

```
/*
```

```
Plugin Name: Liton's Nivo Plugin
```

```
Description: Simple implementation of a nivo slideshow into wordpress
```

```
Author: Md. Liton Arefin
```

```
Author URI: http://liton-online.com Version: 1.0
```

```
*/
```

```
?>
```

আমরা প্লাগিনের হেডার তৈরি করলাম।

<input type="checkbox"/>	Liton's Nivo Plugin Deactivate Edit	Simple implementation of a nivo slideshow into wordpress By Md. Liton Arefin
<input type="checkbox"/>	Liton Action Deactivate Edit	This is a demo description. Version 1.0 Author: Md. Liton Arefin Visit plugin site

প্লাগিন লিস্টে দেখাচ্ছে ‘Liton's Nivo Plugin’। আমাদের প্রথম কাজ হচ্ছে custom post type ওয়ার্ডপ্রেস ফাংশনালিটি। একটি নতুন কাস্টম পেজ তৈরি করতে custom post type ব্যবহৃত হবে যেখানে একটি ইমেজ থাকবে। আমরা শুধু টাইটেল নেম এবং ইমেজ আপলোডার হিসেবে ফিচার ইমেজ ফাংশনালিটি ব্যবহার করব।

```
function np_init() {
    add_shortcode('np-shortcode', 'np_function');
    add_image_size('np_function', 600, 280, true);
}
```

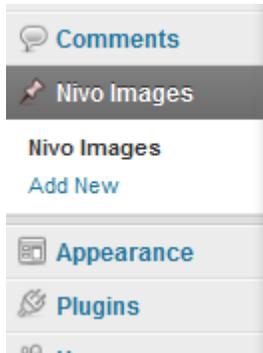
```

    $args = array('public' => true, 'label' => 'Nivo Images', 'supports' =>
array('title', 'thumbnail'));
    register_post_type('np_images', $args);
}
add_action('init', 'np_init');

```

উপরের কোডে np_init() ফাংশনটি হুক করেছি যা হেডার রান করার আগে রান হবে। আর এজন্য register_post_type() ফাংশনকে এর প্যারামিটার দিয়ে np_init() ফাংশনের ভেতরে যুক্ত করেছি। এটা যদি না করতাম তাহলে “Headers already sent!” মেসেজ দেখাত। এই মেসেজটির সাথে অনেকেই পরিচিত আশা করি। register_post_type() ফাংশন দিয়ে আমরা ইমেজ যুক্ত এবং ম্যানেজ করব। মূলত আমরা একটি নতুন custom post type যোগ করেছি np_images নাম দিয়ে।

নতুন custom post type কে সেটিংস ভ্যালু হিসেবে আরগুমেন্ট পাস করেছি। \$args ভেরিয়েবলটি টপ লেভেল মেনু তৈরি করে আগেই দেখেছি। প্লাগিনটি এন্টিভেট করলে আমরা নিচের মত দেখতে পাব-



এবার স্ক্রিপ্ট এবং স্টাইলশীট কে যুক্ত করব-

```

add_action('wp_print_scripts', 'np_register_scripts');
add_action('wp_print_styles', 'np_register_styles');

```

এবার হুক করা ফাংশনগুলো ডিক্লেয়ার করি-

```

function np_register_scripts() {
    if (!is_admin()) {
        // register
        wp_register_script('np_nivo-script', plugins_url('nivo-
slider/jquery.nivo.slider.js', __FILE__), array('jquery' ));
        wp_register_script('np_script', plugins_url('script.js', __FILE__));

        // enqueue
        wp_enqueue_script('np_nivo-script');
        wp_enqueue_script('np_script');
    }
}

```

```

    }
}

function np_register_styles() {
    // register
    wp_register_style('np_styles', plugins_url('nivo-slider/nivo-slider.css',
    __FILE__));
    wp_register_style('np_styles_theme', plugins_url('nivo-
    slider/themes/default/default.css', __FILE__));

    // enqueue
    wp_enqueue_style('np_styles');
    wp_enqueue_style('np_styles_theme');
}

```

উপরের কোডগুলো নতুন করে বর্ণনা করার কিছু নেই। আগেই wp_enqueue_style() এবং wp_enqueue_script() সম্পর্কে বলা হয়েছে।

নতুন আরেকটি ফাইল তৈরি করুন একই ডিরেক্টরিতে script.js এবং নিচের কোডটুকু দিন-
jQuery(document).ready(function(\$) {

```

    $('#slider').nivoSlider();
});

```

ইমেজকে রিসাইজ করুন-

```

add_image_size('np_widget', 180, 100, true);
add_image_size('np_function', 600, 280, true);

```

আমরা এই প্লাগিনটিকে উইজেট হিসেবেও ব্যবহার করব, সেটি পরে দেখব। আরেকটি জিনিস যদি থিমে না দেয়া থাকে সেজন্য থাম্বনেইল সাপোর্ট দিয়ে দিন-

```

add_theme_support( 'post-thumbnails' );

```

এবার ওয়ার্ডপ্রেসের কারেন্ট পোস্ট কুয়েরী করার কোড আর \$result ভেরিয়েবলে যেসব <div> ট্যাগ ব্যবহার করা হয়েছে তা jQuery Plugin টি একত্রীকৃত করে index.html পেজটির মেইন স্লাইডার কোডটি দেখুন, একই আছে। শুধু সেখানকার <div> ট্যাগ গুলোকে কপি করে এই প্লাগিনে ব্যবহার করা হয়েছে।

এবার সর্টকোড দিন-

```

add_shortcode('np-shortcode', 'np_function');

```

এই সর্টকোড পিএইচপি ফাংশনকে হুক করবে এবং এর মধ্য থেকে স্লাইডশোর কনটেন্ট কে রিটার্ন করবে।

আমাদের সর্টকোড হচ্ছে [np-shortcode]

আমরা প্লাগিনে যাই এবং একটি ছবি যুক্ত করি ফিচারড ইমেজ হিসেবে। নিচে দেখুন-

Liton Arefin

Permalink: http://localhost/dev-liton/?np_images=liton-arefin[Change Permalinks](#)[View Post](#)

Publish

[Preview Changes](#)Status: **Published** [Edit](#)Visibility: **Public** [Edit](#)Published on: **Aug 27, 2012 @ 21:46** [Edit](#)[Move to Trash](#)[Update](#)

Featured Image

[Remove featured image](#)

আমি আরেকটি পেজ তৈরি করেছি যেখানে সর্টকোড দিয়েছি

Slider

Permalink: http://localhost/dev-liton/?page_id=2[Change Permalinks](#)[View Page](#)

Upload/Insert



[np-shortcode]

আউটপুট পেজ দেখি এবার-

Edit

Slider



হয়ে গেল আমাদের স্লাইডার প্লাগিন। পুরো কোডটি নিচে দিলাম-

```
<?php
/*
```

```
Plugin Name: Liton's Nivo Plugin
```

```
Description: Simple implementation of a nivo slideshow into wordpress
```

```
Author: Md. Liton Arefin
```

```
Author URI: http://liton-online.com
```

```
*/
```

```
function np_register_scripts() {
    if (!is_admin()) {
        // register
        wp_register_script('np_nivo-script', plugins_url('nivo-
slider/jquery.nivo.slider.js', __FILE__));
        wp_register_script('np_script', plugins_url('script.js', __FILE__));
        // enqueue
        wp_enqueue_script('jquery');
        wp_enqueue_script('np_nivo-script');
        wp_enqueue_script('np_script');
    }
}
```

```
function np_register_styles() {
```

```

// register
wp_register_style('np_styles', plugins_url('nivo-slider/nivo-slider.css',
__FILE__));
wp_register_style('np_styles_theme', plugins_url('nivo-
slider/themes/default/default.css', __FILE__));

// enqueue

wp_enqueue_style('np_styles');
wp_enqueue_style('np_styles_theme');
}

/* function np_widgets_init() {
register_widget('np_Widget');
} */

function np_function($type='np_function') {
    $args = array('post_type' => 'np_images', 'posts_per_page' => 5);
    $result = '<div class="slider-wrapper theme-default">';
    $result .= '<div id="slider" class="nivoSlider">';
    //the loop
    $loop = new WP_Query($args);
    while ($loop->have_posts()) {
        $loop->the_post();

        $the_url = wp_get_attachment_image_src(get_post_thumbnail_id($post-
>ID), $type);

        $result .= '';
    }
    $result .= '</div>';
    $result .= '<div id = "htmlcaption" class = "nivo-html-caption">';
    $result .= '<strong>This</strong> is an example of a <em>HTML</em>
caption with <a href = "#">a link</a>.';
    $result .= '</div>';
    $result .= '</div>';
    return $result;
}

```

```

function np_init() {
    add_shortcode('np_shortcode', 'np_function');
    // add_image_size('np_widget', 180, 100, true);
    add_image_size('np_function', 600, 280, true);

    $args = array('public' => true, 'label' => 'Nivo Images', 'supports' =>
array('title', 'thumbnail'));
    register_post_type('np_images', $args);
}

//hooks
add_theme_support('post-thumbnails');
add_action('init', 'np_init');
add_action('wp_print_scripts', 'np_register_scripts');
add_action('wp_print_styles', 'np_register_styles');
?>

```

এবার আপনি যদি উইজেট হিসেবে ব্যবহার করতে চান তাহলে নিচের কোডটুকু দিন-

```

class np_Widget extends WP_Widget {

    public function __construct() {
        parent::__construct('np_Widget', 'Nivo Slideshow', array('description' =>
__('A Nivo Slideshow Widget', 'text_domain')));
    }

    public function form($instance) {
        if (isset($instance['title'])) {
            $title = $instance['title'];
        } else {
            $title = __('Widget Slideshow', 'text_domain');
        }
        ?>
        <p>
            <label for="<?php echo $this->get_field_id('title'); ?>"><?php
_e('Title:'); ?></label>

```

```

        <input class="widefat" id="<?php echo $this->get_field_id('title'); ?>"
name="<?php echo $this->get_field_name('title'); ?>" type="text" value="<?php
echo esc_attr($title); ?>" />
    </p>
    <?php
}

```

```

public function update($new_instance, $old_instance) {
    $instance = array();
    $instance['title'] = strip_tags($new_instance['title']);

    return $instance;
}
public function widget($args, $instance) {
    extract($args);
    //the title
    $title = apply_filters('widget_title', $instance['title']);
    echo $before_widget;
    if (!empty($title))
        echo $before_title . $title . $after_title;
    echo np_function('np_widget');
    echo $after_widget;
}
}

```

আরো দুটি কাজ করতে হবে আপনাকে। এক, ইমেজ সাইজ উইজেট এর জন্য ঠিক করে দিতে হবে-

```

add_image_size('np_widget', 180, 100, true);
আরেকটি হল উইজেট এ্যাকশন হুক যোগ করা।
add_action('widgets_init', 'np_widgets_init');

```

উইজেট সহ পুরো কোড আবার দিলাম-

```

<?php
/*

```

Plugin Name: Nivo Plugin

Description: Simple implementation of a nivo slideshow into wordpress

Author: Ciprian Turcu

Version: 1.0

```

*/

class np_Widget extends WP_Widget {

    public function __construct() {
        parent::__construct('np_Widget', 'Nivo Slideshow', array('description' =>
__('A Nivo Slideshow Widget', 'text_domain')));
    }

    public function form($instance) {
        if (isset($instance['title'])) {
            $title = $instance['title'];
        } else {
            $title = __('Widget Slideshow', 'text_domain');
        }
        ?>
        <p>
            <label for="<?php echo $this->get_field_id('title'); ?>"><?php
__e('Title:'); ?></label>
            <input class="widefat" id="<?php echo $this->get_field_id('title'); ?>"
name="<?php echo $this->get_field_name('title'); ?>" type="text" value="<?php
echo esc_attr($title); ?>" />
        </p>
        <?php
    }

    public function update($new_instance, $old_instance) {
        $instance = array();
        $instance['title'] = strip_tags($new_instance['title']);

        return $instance;
    }

    public function widget($args, $instance) {
        extract($args);
        //the title
        $title = apply_filters('widget_title', $instance['title']);
        echo $before_widget;
        if (!empty($title))
            echo $before_title . $title . $after_title;
        echo np_function('np_widget');
    }
}

```

```

        echo $after_widget;
    }

}

function np_register_scripts() {
    if (!is_admin()) {
        // register
        wp_register_script('np_nivo-script', plugins_url('nivo-
slider/jquery.nivo.slider.js', __FILE__));
        wp_register_script('np_script', plugins_url('script.js', __FILE__));
        // enqueue
        wp_enqueue_script('jquery');
        wp_enqueue_script('np_nivo-script');
        wp_enqueue_script('np_script');
    }
}

function np_register_styles() {
    // register
    wp_register_style('np_styles', plugins_url('nivo-slider/nivo-slider.css',
__FILE__));
    wp_register_style('np_styles_theme', plugins_url('nivo-
slider/themes/default/default.css', __FILE__));

    // enqueue

    wp_enqueue_style('np_styles');
    wp_enqueue_style('np_styles_theme');
}

function np_widgets_init() {
    register_widget('np_Widget');
}

function np_function($type='np_function') {
    $args = array('post_type' => 'np_images', 'posts_per_page' => 5);
    $result = '<div class="slider-wrapper theme-default">';
    $result .= '<div id="slider" class="nivoSlider">';
    //the loop

```

```

$loop = new WP_Query($args);
while ($loop->have_posts()) {
    $loop->the_post();

    $the_url = wp_get_attachment_image_src(get_post_thumbnail_id($post-
>ID), $type);

    $result .= '';
    }
    $result .= '</div>';
    $result .= '<div id = "htmlcaption" class = "nivo-html-caption">';
    $result .= '<strong>This</strong> is an example of a <em>HTML</em>
caption with <a href = "#">a link</a>.';
    $result .= '</div>';
    $result .= '</div>';
    return $result;
}

function np_init() {
    add_shortcode('np-shortcode', 'np_function');

    add_image_size('np_widget', 180, 100, true);
    add_image_size('np_function', 600, 280, true);

    $args = array('public' => true, 'label' => 'Nivo Images', 'supports' =>
array('title', 'thumbnail'));
    register_post_type('np_images', $args);
}

//hooks
add_theme_support('post-thumbnails');
add_action('init', 'np_init');
add_action('widgets_init', 'np_widgets_init');
add_action('wp_print_scripts', 'np_register_scripts');
add_action('wp_print_styles', 'np_register_styles');
?>

```

প্লাগিন নিয়ে এত ছোট পরিসরে লেখা সম্ভব নয়, এজন্যই মনে হয় প্লাগিন নিয়ে কেউ লেখেন না। কারণ এখানে এত বিষয় আছে যা আলোচনা করতে গেলে ৫০০ পেজ পার হয়ে যাবে। বোঝার সুবিধার্থে আমি যথাসম্ভব উদাহরন দিয়ে বোঝানোর চেষ্টা করেছি। আমি এখানে টার্গেট করেছি যারা টুকটাক জানে প্লাগিন সম্পর্কে, এডভান্স লেভেলের নয় কারণ তারা জানেন। আমার উদ্দেশ্য ছিল যারা জানতে চায় তাদেরকে জানানো। এখানে যেসব বিষয় আলোচনা করা হয়েছে, তা মোটামুটি মানের একটি প্লাগিন বানানোর জন্য যথেষ্ট। তবে মানোন্নয়নের জন্য আপনাকে নিজেই খুঁজে নিতে হবে আপনার জন্য কোনটা ভাল হবে। আমি শুধু দিক নির্দেশনা দিলাম মাত্র। এই বইটি শুধুমাত্র ওয়ার্ডপ্রেস গ্রুপের জন্য লেখা। অনেক ভুল ভ্রান্তি থাকতে পারে। আপনারা ক্ষমাসুন্দর দৃষ্টিতে দেখবেন। আমি চেষ্টা করেছি যতটা সহজ করে লেখা যায়। অনেক সময় দেখা যায় যে খুব ছোট খাটো বিষয় নিয়ে অনেকেই নক করেন, এটা সত্যি দুঃখজনক। আপনারা গুগলে একটু খুঁজলেই সমাধান পাবেন। আগে নিজে চেষ্টা করুন তার পরে অন্যকে সমাধানের জন্য জিজ্ঞাসা করুন। আমার জন্য দোয়া করবেন। ভালো লাগলে জানাবেন, খারাপ লাগলেও জানাবেন। তাহলেই আমার পরিশ্রম স্বার্থক হবে। মতামত জানিয়ে আমাকে মেইল করতে পারেন litonice09@gmail.com |

Dedicated
To
My Parents