

DAILY ASSESSMENT FORMAT

Date:	11/06/2020	Name:	Nishanth
Course:	Pcb design	USN:	4a117ec063
Topic:	1.create pcb footprint component 2.add footprint search path	Semester & Section:	6thb-section
GitHub Repository:	nishanthvr		

FORENOON SESSION DETAILS

Creating a Custom KiCad Footprint Library

This section will show you how to create your own local custom footprints so that you can use them and connect them to schematic components using CvPcb. We're going to assume you've already been through the previous sections of this tutorial; you should have KiCad downloaded and installed.

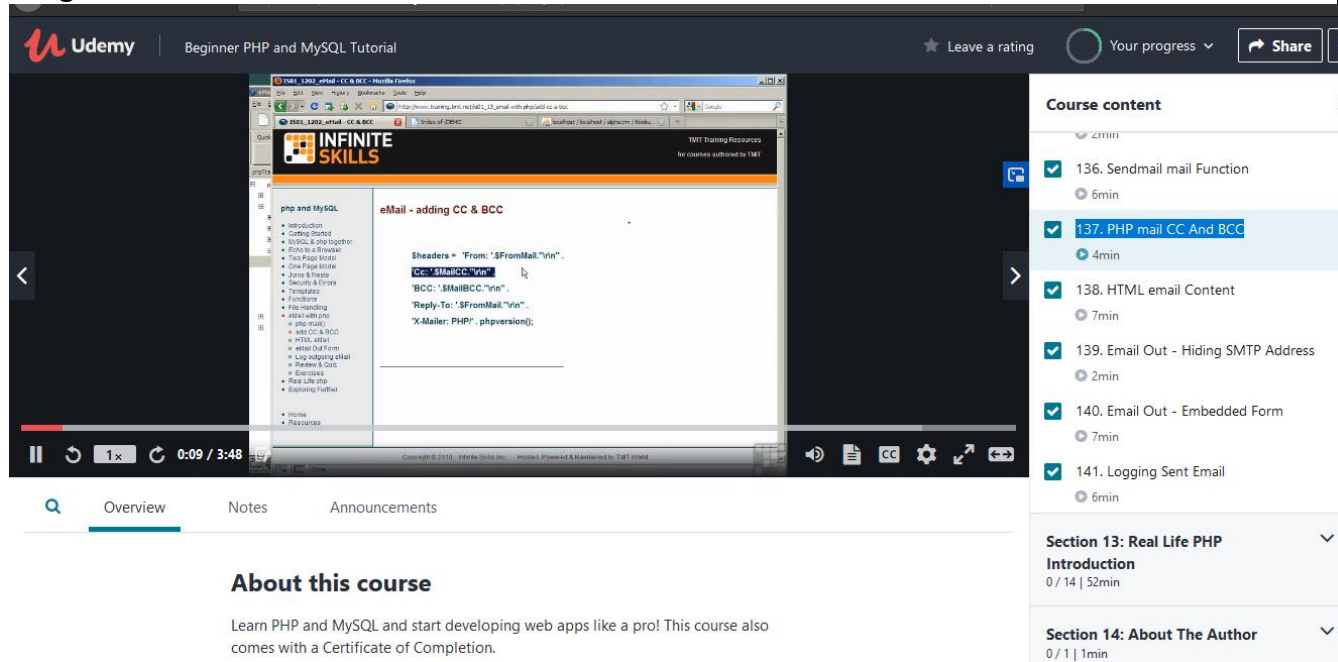
Open KiCad's project manager and then click on the PCB footprint editor button

1. Search for desired part and download from SnapEDA.
2. In KiCad, go to Tools - Run EESchema.
3. Select Preferences - Component Libraries.
4. In the Component library files, click Add.
5. Select the .lib library files.
6. Go to Tools - Open PcbNew.
7. Click - Footprint Libraries

Date: 10/06/2020
Course: Beginner PHP and MySQL
Name: Nishanth
USN: 4a17ec063
Semester & Section: 6th and b section
1.php mail cc and bcc
2.html email content
3.email out hiding smtp address

AFTERNOON SESSION DETAILS

Image of session



Course content

- 136. Sendmail mail Function (6min)
- 137. PHP mail CC And BCC (4min)
- 138. HTML email Content (7min)
- 139. Email Out - Hiding SMTP Address (2min)
- 140. Email Out - Embedded Form (7min)
- 141. Logging Sent Email (6min)

Section 13: Real Life PHP Introduction
0 / 14 | 52min

Section 14: About The Author
0 / 1 | 1min

About this course

Learn PHP and MySQL and start developing web apps like a pro! This course also comes with a Certificate of Completion.

Php gmail content:

```
<?php
require __DIR__ . '/vendor/autoload.php';

if (php_sapi_name() != 'cli') {
    throw new Exception('This application must be run on the command line.');
```

/**

- * Returns an authorized API client.
- * @return Google_Client the authorized client object

*/

```
function getClient()
{
    $client = new Google_Client();
    $client->setApplicationName('Gmail API PHP Quickstart');
    $client->setScopes(Google_Service_Gmail::GMAIL_READONLY);
    $client->setAuthConfig('credentials.json');
    $client->setAccessType('offline');
    $client->setPrompt('select_account consent');
```

// Load previously authorized token from a file, if it exists.
// The file token.json stores the user's access and refresh tokens, and is
// created automatically when the authorization flow completes for the first
// time.

```
$tokenPath = 'token.json';
```

```

    if (file_exists($tokenPath)) {
        $accessToken = json_decode(file_get_contents($tokenPath), true);
        $client->setAccessToken($accessToken);
    }

    // If there is no previous token or it's expired.
    if ($client->isAccessTokenExpired()) {
        // Refresh the token if possible, else fetch a new one.
        if ($client->getRefreshToken()) {
            $client->fetchAccessTokenWithRefreshToken($client->getRefreshToken());
        } else {
            // Request authorization from the user.
            $authUrl = $client->createAuthUrl();
            printf("Open the following link in your browser:\n%s\n", $authUrl);
            print 'Enter verification code: ';
            $authCode = trim(fgets(STDIN));

            // Exchange authorization code for an access token.
            $accessToken = $client->fetchAccessTokenWithAuthCode($authCode);
            $client->setAccessToken($accessToken);

            // Check to see if there was an error.
            if (array_key_exists('error', $accessToken)) {
                throw new Exception(join(', ', $accessToken));
            }
        }
        // Save the token to a file.
        if (!file_exists(dirname($tokenPath))) {
            mkdir(dirname($tokenPath), 0700, true);
        }
        file_put_contents($tokenPath, json_encode($client->getAccessToken()));
    }
    return $client;
}

// Get the API client and construct the service object.
$client = getClient();
$service = new Google_Service_Gmail($client);

// Print the labels in the user's account.
$user = 'me';
$results = $service->users_labels->listUsersLabels($user);

if (count($results->getLabels()) == 0) {
    print "No labels found.\n";
} else {
    print "Labels:\n";
    foreach ($results->getLabels() as $label) {
        printf("- %s\n", $label->getName());
    }
}
}

```