LAMPIRAN – LAMPIRAN

Berikut adalah potongan *source c*ode tampilan halaman *Bibliografi Perpustakaan.*

```
<?php
// key to authenticate
if (!defined('INDEX_AUTH')) {
    define('INDEX_AUTH', '1');
}
#use SLiMS\AdvancedLogging;
use SLiMS\AlLibrarian;
// key to get full database access
define('DB_ACCESS', 'fa');
if (!defined('SB')) {
    // main system configuration
    require '../../../sysconfig.inc.php';
    // start the session
    require SB . 'admin/default/session.inc.php';
}
// IP based access limitation
require LIB . 'ip_based_access.inc.php';
do_checkIP('smc');
do_checkIP('smc-bibliography');
require SB . 'admin/default/session_check.inc.php';
require SIMBIO . 'simbio_GUI/table/simbio_table.inc.php';
require SIMBIO . 'simbio_GUI/form_maker/simbio_form_table_AJAX.inc.php';
require SIMBIO . 'simbio_GUI/paging/simbio_paging.inc.php';
require SIMBIO . 'simbio_DB/datagrid/simbio_dbgrid.inc.php';
require SIMBIO . 'simbio_DB/simbio_dbop.inc.php';
require SIMBIO . 'simbio_FILE/simbio_file_upload.inc.php';
require MDLBS . 'system/biblio_indexer.inc.php';

// privileges checking
$can_read = utility::havePrivilege('bibliography', 'r');
$can_write = utility::havePrivilege('bibliography', 'w');
if (!$can_read) {
    die('<div class="errorBox">' . __('You are not authorized to view this section') . '</div>');
}
// execute registered hook
\SLiMS\Plugins::getInstance()->execute('bibliography_init');
// load settings
utility::loadSettings($dbs);
$in_pop_up = false;
// check if we are inside pop-up window
if (isset($_GET['inPopUp'])) {
    $in_pop_up = true;
}
if (!function_exists('getimagesizefromstring')) {
    function getimagesizefromstring($string_data)
    {
        $uri = 'data://application/octet-stream;base64,' . base64_encode($string_data);
        return getimagesize($uri);
    }
}
// RDA Content, Media and Carrier
$rda_cmc = array('content' => 'Content Type', 'media' => 'Media Type', 'carrier' => 'Carrier Type');
/* REMOVE IMAGE */
if (isset($_POST['removeImage']) && isset($_POST['bimg']) && isset($_POST['img'])) {
    // validate post image
    $biblio_id = utility::filterData('bimg', 'post', true, true, true);
    $image_name = utility::filterData('img', 'post', true, true, true);
    $query_image = $dbs->query("SELECT biblio_id FROM biblio WHERE biblio_id='{$biblio_id}' AND
image='{$image_name}'");
    if ($query_image->num_rows > 0) {
        $_delete = $dbs->query(sprintf('UPDATE biblio SET image=NULL WHERE biblio_id=%d', $biblio_id));
        $_delete2 = $dbs->query(sprintf('UPDATE search_biblio SET image=NULL WHERE biblio_id=%d',
$biblio_id));
        if ($_delete) {
            $postImage = stripslashes($_POST['img']);
            $postImage = str_replace('/', '', $postImage);
            @unlink(sprintf(IMGBS . 'docs/%s', $postImage));
            utility::jsToastr('Bibliography', str_replace('{imageFilename}', $_POST['img'], __('{imageFilename}
successfully removed!')), 'success');
            // exit('<script type="text/javascript">$(\'#biblioImage, #imageFilename\').remove();</script>');
            exit('<img src="../lib/minigalnano/createthumb.php?filename=images/default/image.png&width=130"
class="img-fluid rounded" alt="">');
        }
    }
    exit();
}
/* RECORD OPERATION */
if (isset($_POST['saveData']) AND $can_read AND $can_write) {
    if (!simbio_form_maker::isTokenValid()) {
        utility::jsToastr('Bibliography', __('Invalid form submission token!'), 'error');
        utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'system', 'Invalid form submission token, might be a CSRF
attack from ' . $_SERVER['REMOTE_ADDR']);
        exit();
    }
    $title = trim(strip_tags($_POST['title']));
    // check form validity
    if (empty($title)) {
        utility::jsToastr('Bibliography', __('Title can not be empty'), 'error');
        exit();
    } else {
        // include custom fields file
        if (file_exists(MDLBS . 'bibliography/custom_fields.inc.php')) {
            include MDLBS . 'bibliography/custom_fields.inc.php';
        }
        // create biblio_indexer class instance
        $indexer = new biblio_indexer($dbs);
        /**
         * Custom fields
         */
        if (isset($biblio_custom_fields)) {
            if (is_array($biblio_custom_fields) && $biblio_custom_fields) {
                foreach ($biblio_custom_fields as $fid => $cfield) {
                    // custom field data
                    $cf_dbfield = $cfield['dbfield'];
                    if (isset($_POST[$cf_dbfield])) {
                        if (is_array($_POST[$cf_dbfield])) {
                            foreach ($_POST[$cf_dbfield] as $value) {
                                $arr[$value] = $value;
                            }
                            $custom_data[$cf_dbfield] = serialize($arr);
                        } else {
                            $cf_val = $dbs->escape_string(strip_tags(trim($_POST[$cf_dbfield]),
$sysconf['content']['allowable_tags']));
                            if ($cfield['type'] == 'numeric' && (!is_numeric($cf_val) && $cf_val != '')) {
                                utility::jsToastr(__('Bibliography'), sprintf(__('Field %s only number for allowed'), $cfield['label']),
'error');
                                exit();
                            } elseif ($cfield['type'] == 'date' && $cf_val == '') {
```

```
                        utility::jsToastr(__('Bibliography'), sprintf(__('Field %s is date format, empty not allowed'),
$cfield['label']), 'error');
                        exit();
                    }
                    $custom_data[$cf_dbfield] = $cf_val;
                }
            } else {
                $custom_data[$cf_dbfield] = serialize(array());
            }
        }
    }
}
$data['title'] = $dbs->escape_string($title);
/* modified by hendro */
$data['sor'] = trim($dbs->escape_string(strip_tags($_POST['sor'])));
/* end of modification */
$data['edition'] = trim($dbs->escape_string(strip_tags($_POST['edition'])));
$data['gmd_id'] = $_POST['gmdID'];
$data['isbn_issn'] = trim($dbs->escape_string(strip_tags($_POST['isbn_issn'])));
$class = str_ireplace('NEW:', '', trim(strip_tags($_POST['class'])));
$data['classification'] = trim($dbs->escape_string(strip_tags($class)));
$data['uid'] = $_SESSION['uid'];
// check publisher
// echo stripos($_POST['publisherID'], 'NEW:');
if (stripos($_POST['publisherID'], 'NEW:') === 0) {
    $new_publisher = str_ireplace('NEW:', '', trim(strip_tags($_POST['publisherID'])));
    $new_id = utility::getID($dbs, 'mst_publisher', 'publisher_id', 'publisher_name', $new_publisher);
    $data['publisher_id'] = $new_id;
} else if (intval($_POST['publisherID']) > 0) {
    $data['publisher_id'] = intval($_POST['publisherID']);
}
$data['publish_year'] = trim($dbs->escape_string(strip_tags($_POST['year'])));
$data['collation'] = trim($dbs->escape_string(strip_tags($_POST['collation'])));
$data['series_title'] = trim($dbs->escape_string(strip_tags($_POST['seriesTitle'])));
$data['call_number'] = trim($dbs->escape_string(strip_tags($_POST['callNumber'])));
$data['language_id'] = trim($dbs->escape_string(strip_tags($_POST['languageID'])));
// check place
if (stripos($_POST['placeID'], 'NEW:') === 0) {
    $new_place = str_ireplace('NEW:', '', trim(strip_tags($_POST['placeID'])));
    $new_id = utility::getID($dbs, 'mst_place', 'place_id', 'place_name', $new_place);
    $data['publish_place_id'] = $new_id;
} else if (intval($_POST['placeID']) > 0) {
    $data['publish_place_id'] = intval($_POST['placeID']);
}
$data['notes'] = trim($dbs->escape_string(strip_tags($_POST['notes'],
'<br><p><div><span><i><em><strong><b><code>s')));
$data['opac_hide'] = ($_POST['opacHide'] == '0') ? 'literal{0}' : '1';
$data['promoted'] = ($_POST['promote'] == '0') ? 'literal{0}' : '1';
// labels
$arr_label = array();
if (!empty($_POST['labels'])) {
    foreach ($_POST['labels'] as $label) {
        if (trim($label) != '') {
            $arr_label[] = array($label, isset($_POST['label_urls'][$label]) ? $_POST['label_urls'][$label] : null);
        }
    }
}
$data['labels'] = $arr_label ? serialize($arr_label) : 'literal{NULL}';
$data['frequency_id'] = ($_POST['frequencyID'] == '0') ? 'literal{0}' : (integer)$_POST['frequencyID'];
$data['spec_detail_info'] = trim($dbs->escape_string(strip_tags($_POST['specDetailInfo'])));
// RDA Content, Media anda Carrier Type
foreach ($rda_cmc as $cmc => $cmc_name) {
        if (isset($_POST[$cmc . 'TypeID']) && $_POST[$cmc . 'TypeID'] <> 0) {
          $data[$cmc . '_type_id'] = filter_input(INPUT_POST, $cmc . 'TypeID',
FILTER_SANITIZE_NUMBER_INT);
        }
      }

      $data['input_date'] = date('Y-m-d H:i:s');
      $data['last_update'] = date('Y-m-d H:i:s');
      // image uploading
      if (!empty($_FILES['image']) AND $_FILES['image']['size']) {
        // create upload object
        $image_upload = new simbio_file_upload();
        $image_upload->setAllowableFormat($sysconf['allowed_images']);
        $image_upload->setMaxSize($sysconf['max_image_upload'] * 1024);
        $image_upload->setUploadDir(IMGBS . 'docs');
        // upload the file and change all space characters to underscore
        $img_upload_status = $image_upload->doUpload('image', preg_replace('@\s+@i', '_',
$_FILES['image']['name']));
        if ($img_upload_status == UPLOAD_SUCCESS) {
          $data['image'] = $dbs->escape_string($image_upload->new_filename);
          // write log
          utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'bibliography', $_SESSION['realname'] . ' upload image
file ' . $image_upload->new_filename);
          utility::jsToastr('Bibliography', __('Image Uploaded Successfully'), 'success');
        } else {
          // write log
          utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'bibliography', 'ERROR : ' . $_SESSION['realname'] . '
FAILED TO upload image file ' . $image_upload->new_filename . ', with error (' . $image_upload->error . ')');
          utility::jsToastr('Bibliography', __('Image Uploaded Failed'), 'error');
        }
      } else if (!empty($_POST['base64picstring'])) {
        list($filedata, $filedom) = explode('#image/type#', $_POST['base64picstring']);
        $filedata = base64_decode($filedata);
        $fileinfo = getimagesizefromstring($filedata);
        $valid = strlen($filedata) / 1024 < $sysconf['max_image_upload'];
        $valid = (!$fileinfo || $valid === false) ? false : in_array($fileinfo['mime'],
$sysconf['allowed_images_mimetype']);
        $new_filename = strtolower('cover_'
          . preg_replace("/[^a-zA-Z0-9]+/", "_", $data['title'])
          . '.' . $filedom);
        if ($valid AND file_put_contents(IMGBS . 'docs/' . $new_filename, $filedata)) {
          $data['image'] = $dbs->escape_string($new_filename);
          if (!defined('UPLOAD_SUCCESS')) define('UPLOAD_SUCCESS', 1);
          $upload_status = UPLOAD_SUCCESS;
        }
      }
      // create sql op object
      $sql_op = new simbio_dbop($dbs);
      if (isset($_POST['updateRecordID'])) {
        if ($sysconf['log']['biblio']) {
          $_prevrawdata = api::biblio_load($dbs, $_POST['updateRecordID']);
        }
        /* UPDATE RECORD MODE */
        // remove input date
        unset($data['input_date']);
        unset($data['uid']);
        // filter update record ID
        $updateRecordID = (integer)$_POST['updateRecordID'];
        \SLiMS\Plugins::getInstance()->execute('bibliography_before_update', ['data' => array_merge($data,
['biblio_id' => $updateRecordID])]);
        // update data
        $update = $sql_op->update('biblio', $data, 'biblio_id=' . $updateRecordID);
```

```
          // send an alert
          if ($update) {
             // execute registered hook
             \SLiMS\Plugins::getInstance()->execute('bibliography_after_update', ['data' => array_merge($data,
['biblio_id' => $updateRecordID])]);
             // update custom data
             if (isset($custom_data)) {
                // check if custom data for this record exists
                $_sql_check_custom_q = sprintf('SELECT biblio_id FROM biblio_custom WHERE biblio_id=%d',
$updateRecordID);
                $check_custom_q = $dbs->query($_sql_check_custom_q);
                if ($check_custom_q->num_rows) {
                   $update2 = @$sql_op->update('biblio_custom', $custom_data, 'biblio_id=' . $updateRecordID);
                } else {
                   $custom_data['biblio_id'] = $updateRecordID;
                   @$sql_op->insert('biblio_custom', $custom_data);
                }
             }
             if ($sysconf['bibliography_update_notification']) {
                utility::jsToastr('Bibliography', __('Bibliography Data Successfully Updated'), 'success');
             }
             // auto insert catalog to UCS if enabled
             if ($sysconf['ucs']['enable']) {
                echo '<script type="text/javascript">parent.ucsUpload(\'' . MWB . 'bibliography/ucs_upload.php\',
\'itemID[]=' . $updateRecordID . '\', false);</script>';
             }
             // write log
             utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'bibliography', $_SESSION['realname'] . ' update
bibliographic data (' . $data['title'] . ') with biblio_id (' . $updateRecordID . ')');
             if ($sysconf['log']['biblio']) {
                $_currrawdata = api::biblio_load($dbs, $updateRecordID);
                api::bibliolog_compare($dbs, $updateRecordID, $_SESSION['uid'], $_SESSION['realname'],
$data['title'], $_currrawdata, $_SESSION['_prevrawdata'][$updateRecordID]);
                unset($_SESSION['_prevrawdata'][$updateRecordID]);
             }
             if ($sysconf['index']['engine']['enable']) {
                api::update_to_index($_currrawdata);
             }
             // close window OR redirect main page
             if ($in_pop_up) {
                $itemCollID = (integer)$_POST['itemCollID'];
                echo '<script
type="text/javascript">top.$(\'#mainContent\').simbioAJAX(parent.jQuery.ajaxHistory[0].url, {method: \'post\',
addData: \'' . ($itemCollID ? 'itemID=' . $itemCollID . '&detail=true' : '') . '\'});</script>';
                echo '<script type="text/javascript">top.closeHTMLpop();</script>';
             } else {
                echo '<script
type="text/javascript">top.$(\'#mainContent\').simbioAJAX(parent.jQuery.ajaxHistory[0].url);</script>';
             }
             // update index
             // delete from index first
             $sql_op->delete('search_biblio', "biblio_id=$updateRecordID");
             $indexer->makeIndex($updateRecordID);
          } else {
             utility::jsToastr('Bibliography', __('Bibliography Data FAILED to Updated. Please Contact System
Administrator') . "\n" . $sql_op->error, 'error');
          }
       } else {
          // execute registered hook
          \SLiMS\Plugins::getInstance()->execute('bibliography_before_save', ['data' => $data]);
          /* INSERT RECORD MODE */
          // insert the data
```

```
            $insert = $sql_op->insert('biblio', $data);
            if ($insert) {
                // get auto id of this record
                $last_biblio_id = $sql_op->insert_id;
                // execute registered hook
                $data['biblio_id'] = $last_biblio_id;
                \SLiMS\Plugins::getInstance()->execute('bibliography_after_save', ['data' => $data]);
                // add authors
                if ($_SESSION['biblioAuthor']) {
                    foreach ($_SESSION['biblioAuthor'] as $author) {
                        $sql_op->insert('biblio_author', array('biblio_id' => $last_biblio_id, 'author_id' => $author[0], 'level' =>
$author[1]));
                    }
                }
                // add topics
                if ($_SESSION['biblioTopic']) {
                    foreach ($_SESSION['biblioTopic'] as $topic) {
                        $sql_op->insert('biblio_topic', array('biblio_id' => $last_biblio_id, 'topic_id' => $topic[0], 'level' =>
$topic[1]));
                    }
                }
                // add attachment
                if ($_SESSION['biblioAttach']) {
                    foreach ($_SESSION['biblioAttach'] as $attachment) {
                        $sql_op->insert('biblio_attachment', array('biblio_id' => $last_biblio_id, 'file_id' =>
$attachment['file_id'], 'access_type' => $attachment['access_type']));
                    }
                }
                // biblio to biblio
                if ($_SESSION['biblioToBiblio']) {
                    foreach ($_SESSION['biblioToBiblio'] as $rel_biblio_id) {
                        $sql_op->insert('biblio_relation', array('biblio_id' => $last_biblio_id, 'rel_biblio_id' =>
$rel_biblio_id[0]));
                    }
                }
                // insert custom data
                if (isset($custom_data)) {
                    $custom_data['biblio_id'] = $last_biblio_id;
                    @$sql_op->insert('biblio_custom', $custom_data);
                }
                utility::jsToastr('Bibliography', __('New Bibliography Data Successfully Saved'), 'success');
                // write log
                utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'bibliography', $_SESSION['realname'] . ' insert
bibliographic data (' . $data['title'] . ') with biblio_id (' . $last_biblio_id . ')');
                if ($sysconf['log']['biblio']) {
                    $_rawdata = api::biblio_load($dbs, $last_biblio_id);
                    api::bibliolog_write($dbs, $last_biblio_id, $_SESSION['uid'], $_SESSION['realname'], $data['title'],
'create', 'description', $_rawdata, 'New data. Bibliography.');
                    api::bibliolog_compare($dbs, $last_biblio_id, $_SESSION['uid'], $_SESSION['realname'], $data['title'],
$_rawdata, NULL);
                }
                if ($sysconf['index']['engine']['enable']) {
                    api::update_to_index($_rawdata);
                }
                // clear related sessions
                $_SESSION['biblioAuthor'] = array();
                $_SESSION['biblioTopic'] = array();
                $_SESSION['biblioAttach'] = array();
                $_SESSION['biblioToBiblio'] = array();
                // update index
                $indexer->makeIndex($last_biblio_id);
                // auto insert catalog to UCS if enabled
            if ($sysconf['ucs']['enable'] && $sysconf['ucs']['auto_insert']) {
                echo '<script type="text/javascript">parent.ucsUpload(\'' . MWB . 'bibliography/ucs_upload.php\',
\'itemID[]=' . $last_biblio_id . '\');</script>';
            }
        } else {
            utility::jsToastr('Bibliography', __('Bibliography Data FAILED to Save. Please Contact System
Administrator') . "\n" . $sql_op->error, 'error');
        }
    }
    // item batch insert
    if (trim($_POST['itemCodePattern']) != '' && $_POST['totalItems'] > 0) {
        $pattern = trim($_POST['itemCodePattern']);
        $total = (integer)$_POST['totalItems'];
        $regex = '/0{3,}/';
        if ($total > $sysconf['max_insert_batch']) {
            utility::jsToastr('Bibliography', sprintf(__('Item Data FAILED to Save. Insert batch item maximum %s
copies'), $sysconf['max_insert_batch']), 'warning');
            die();
        }
        // get zeros
        preg_match($regex, $pattern, $result);
        $zeros = strlen($result[0]);
        // get chars
        $chars = preg_split($regex, $pattern);
        $chars_last = (isset($chars[1]) && !empty(trim($chars[1]))) ? trim($chars[1]) : '';
        // get last number from database
        $last_q = $dbs->query('SELECT item_code FROM item WHERE item_code REGEXP \'^' . $chars[0] . '[0-
9]{3,}' . $chars_last . '$\' ORDER BY item_code DESC LIMIT 1');
        if (!$dbs->errno && $last_q->num_rows > 0) {
            $last_d = $last_q->fetch_row();
            // get last  number
            $ptn = '/' . $chars[0] . '|' . $chars_last . '$/';
            $last = preg_replace($ptn, '', $last_d[0]);
            $start = intval($last) + 1;
        } else {
            $start = 1;
        }
        $end = $start + $total;
        for ($b = $start; $b < $end; $b++) {
            $len = strlen($b);
            $itemcode = $chars[0];
            if ($zeros > 0) {
                $itemcode .= preg_replace('@0{' . $len . '}$@i', $b, $result[0]);
            } else {
                $itemcode .= $b;
            }
            $itemcode .= $chars[1];
            $item_insert_sql = sprintf("INSERT IGNORE INTO item (biblio_id, item_code, received_date, supplier_id,
order_no, order_date, item_status_id, site, source, invoice, price, price_currency, invoice_date, call_number,
coll_type_id, location_id, input_date, last_update, uid)
    VALUES ( %d, '%s', '%s', '%d', '%s', '%s', '%s', '%s', '%s', '%s', %d, '%s', '%s', '%s', '%s', '%s', '%s', '%s',
%d)",
                isset($updateRecordID) ? $updateRecordID : $last_biblio_id, $itemcode, $dbs-
>escape_string($_POST['recvDate']), intval($_POST['supplierID']), $dbs->escape_string($_POST['ordNo']), $dbs-
>escape_string($_POST['orDate']), $dbs->escape_string($_POST['itemStatusID']), $dbs-
>escape_string($_POST['itemSite']), intval($_POST['source']), $dbs->escape_string($_POST['invoice']),
intval($_POST['price']), $dbs->escape_string($_POST['priceCurrency']), $dbs->escape_string($_POST['invcDate']),
$data['call_number'], intval($_POST['collTypeID']), $dbs->escape_string($_POST['locationID']), date('Y-m-d H:i:s'),
date('Y-m-d H:i:s'), $_SESSION['uid']);
            @$dbs->query($item_insert_sql);
        }
    }
```

```
        echo '<script type="text/javascript">parent.$(\'#mainContent\').simbioAJAX(\'' . MWB .
'bibliography/index.php\', {method: \'post\', addData: \'itemID=' . (isset($updateRecordID) ? $updateRecordID :
$last_biblio_id) . '&detail=true\'});</script>';
        exit();
    }
    exit();
} else if (isset($_POST['itemID']) AND !empty($_POST['itemID']) AND isset($_POST['itemAction'])) {
    if (!($can_read AND $can_write)) {
        die();
    }
    if (!simbio_form_maker::isTokenValid()) {
        utility::jsToastr('Bibliography', __('Invalid form submission token!'), 'error');
        utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'system', 'Invalid form submission token, might be a CSRF
attack from ' . $_SERVER['REMOTE_ADDR']);
        exit();
    }
    /* DATA DELETION PROCESS */
    // create sql op object
    $sql_op = new simbio_dbop($dbs);
    $failed_array = array();
    $error_num = 0;
    $still_have_item = array();
    if (!is_array($_POST['itemID'])) {
        // make an array
        $_POST['itemID'] = array((integer)$_POST['itemID']);
    }
    // loop array
    $http_query = '';
    foreach ($_POST['itemID'] as $itemID) {
        $itemID = (integer)$itemID;
        // check if this biblio data still have an item
        $_sql_biblio_item_q = sprintf('SELECT b.title, COUNT(item_id) FROM biblio AS b
      LEFT JOIN item AS i ON b.biblio_id=i.biblio_id
      WHERE b.biblio_id=%d GROUP BY title', $itemID);
        $biblio_item_q = $dbs->query($_sql_biblio_item_q);
        $biblio_item_d = $biblio_item_q->fetch_row();
        if ($biblio_item_d[1] < 1) {
            if ($sysconf['log']['biblio']) {
                $_rawdata = api::biblio_load($dbs, $itemID);
                api::bibliolog_write($dbs, $itemID, $_SESSION['uid'], $_SESSION['realname'], $biblio_item_d[0], 'delete',
'description', $_rawdata, 'Data bibliografi dihapus.');
            }
            if (!$sql_op->delete('biblio', "biblio_id=$itemID")) {
                $error_num++;
            } else {
                // execute registered hook
                \SLiMS\Plugins::getInstance()->execute('bibliography_on_delete', [$itemID]);
                // write log
                utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'bibliography', $_SESSION['realname'] . ' DELETE
bibliographic data (' . $biblio_item_d[0] . ') with biblio_id (' . $itemID . ')');
                // delete related data
                $sql_op->delete('biblio_topic', "biblio_id=$itemID");
                $sql_op->delete('biblio_author', "biblio_id=$itemID");
                $sql_op->delete('biblio_attachment', "biblio_id=$itemID");
                $sql_op->delete('biblio_relation', "biblio_id=$itemID");
                $sql_op->delete('search_biblio', "biblio_id=$itemID");
                // delete serial data
                // check kardex if exist
                $_sql_serial_kardex_q = sprintf('SELECT b.title, COUNT(kardex_id),s.serial_id FROM biblio AS b
          LEFT JOIN `serial` AS s ON b.biblio_id=s.biblio_id
          LEFT JOIN kardex AS k ON s.serial_id=k.serial_id
          WHERE b.biblio_id=%d GROUP BY title', $itemID);
```

```
            $serial_kardex_q = $dbs->query($_sql_serial_kardex_q);
            if ($serial_kardex_q) {
                $serial_kardex_d = $serial_kardex_q->fetch_row();
                // delete kardex
                if ($serial_kardex_d[1] > 1) {
                    $sql_op->delete('kardex', "serial_id=" . $serial_kardex_d[2]);
                }
            }
            //delete serial data
            $sql_op->delete('serial', "biblio_id=$itemID");
            // add to http query for UCS delete
            $http_query .= "itemID[]=$itemID&";
        }
    } else {
        $still_have_item[] = substr($biblio_item_d[0], 0, 45) . '... still have ' . $biblio_item_d[1] . ' copies';
        $error_num++;
    }
}
if ($still_have_item) {
    $titles = '';
    foreach ($still_have_item as $title) {
        $titles .= $title . "\n";
    }
    utility::jsToastr('Bibliography', __('Below data can not be deleted:') . "\n" . $titles, 'error');
    echo '<script type="text/javascript">parent.$(\'#mainContent\').simbioAJAX(\'' . $_SERVER['PHP_SELF'] . '\',
{addData: \'' . $_POST['lastQueryStr'] . '\'});</script>';
    exit();
}
// auto delete data on UCS if enabled
if ($http_query && $sysconf['ucs']['enable'] && $sysconf['ucs']['auto_delete']) {
    echo '<script type="text/javascript">parent.ucsUpdate(\'' . MWB . 'bibliography/ucs_update.php\',
\'nodeOperation=delete&' . $http_query . '\');</script>';
}
// error alerting
if ($error_num == 0) {
    utility::jsToastr('Bibliography', __('All Data Successfully Deleted'), 'success');
    echo '<script type="text/javascript">parent.$(\'#mainContent\').simbioAJAX(\'' . $_SERVER['PHP_SELF'] . '\',
{addData: \'' . $_POST['lastQueryStr'] . '\'});</script>';
} else {
    utility::jsToastr('Bibliography', __('Some or All Data NOT deleted successfully!\nPlease contact system
administrator'), 'warning');
    echo '<script type="text/javascript">parent.$(\'#mainContent\').simbioAJAX(\'' . $_SERVER['PHP_SELF'] . '\',
{addData: \'' . $_POST['lastQueryStr'] . '\'});</script>';
}
exit();
}
/* RECORD OPERATION END */

if (!$in_pop_up) {
    /* search form */
    ?>
    <div class="menuBox">
        <div class="menuBoxInner biblioIcon">
            <div class="per_title">
                <h2><?php echo __('Bibliographic'); ?></h2>
            </div>
            <div class="sub_section">
                <div class="btn-group">
                    <a href="<?php echo MWB; ?>bibliography/index.php"
                       class="btn btn-default"><?php echo __('Bibliographic List'); ?></a>
                    <a href="<?php echo MWB; ?>bibliography/index.php?action=detail"
                       class="btn btn-default"><?php echo __('Add New Bibliography'); ?></a>
```

```
         </div>
         <form name="search" action="<?php echo MWB; ?>bibliography/index.php" id="search" method="get"
            class="form-inline"><?php echo __('Search'); ?>
           <input type="text" name="keywords" id="keywords" class="form-control col-md-3"/>
           <select name="field" class="form-control col-md-2">
             <option value="0"><?php echo __('All Fields'); ?></option>
             <option value="title"><?php echo __('Title/Series Title'); ?> </option>
             <option value="subject"><?php echo __('Topics'); ?></option>
             <option value="author"><?php echo __('Authors'); ?></option>
             <option value="isbn"><?php echo __('ISBN/ISSN'); ?></option>
             <option value="publisher"><?php echo __('Publisher'); ?></option>
           </select>
           <input type="submit" id="doSearch" value="<?php echo __('Search'); ?>"
               class="s-btn btn btn-default"/>
           <div class="btn btn-info" data-toggle="collapse" data-target="#advancedFilter"
              aria-expanded="false"><?php echo __('Advanced Filter'); ?></div>
           <div class="collapse" id="advancedFilter"
              style="padding-top:10px;width:100%; text-align:left !important;">
             <?php echo __('Hide in OPAC'); ?>&nbsp;
             <select name="opac_hide" class="form-control col-md-2">
               <option value=""><?php echo __('ALL'); ?></option>
               <option value="0"><?php echo __('Show'); ?> </option>
               <option value="1"><?php echo __('Hide'); ?></option>
             </select>
             <?php echo __('Promote To Homepage'); ?>&nbsp;
             <select name="promoted" class="form-control col-md-2">
               <option value=""><?php echo __('ALL'); ?></option>
               <option value="0"><?php echo __('Don\'t Promote'); ?> </option>
               <option value="1"><?php echo __('Promote'); ?></option>
             </select>
           </div>
           <?php
           // enable UCS?
           if ($sysconf['ucs']['enable']) {
             ?>
             <a href="#"
               onclick="ucsUpload('<?php echo MWB; ?>bibliography/ucs_upload.php',
serializeChbox('dataList'))"
               class="s-btn btn btn-default notAJAX"><?php echo __('Upload Selected Bibliographic data to Union
Catalog Server*'); ?></a>
             <?php
           }
           ?>
         </form>
       </div>
     </div>
   </div>
   <?php
   /* search form end */
}
/* main content */
if (isset($_GET['action']) && $_GET['action'] == 'history') {
   $biblioID = utility::filterData('biblioID', 'get', true, true, true);
   $table_spec = 'biblio_log AS bl';
   $criteria = 'bl.biblio_id=' . $biblioID;
   // create datagrid
   $datagrid = new simbio_datagrid();
   $datagrid->setSQLColumn('bl.date AS \'' . __('Date') . '\'',
      'bl.realname AS \'' . __('User Name') . '\'',
      'bl.additional_information AS \'' . __('Additional Information') . '\'');
   $datagrid->modifyColumnContent(2, 'callback{affectedDetail}');
   $datagrid->setSQLorder('bl.biblio_log_id DESC');
```

```
    $datagrid->sql_group_by = 'bl.date';
    $datagrid->setSQLCriteria($criteria);
    // set table and table header attributes
    $datagrid->table_attr = 'id="dataList" class="s-table table"';
    $datagrid->table_header_attr = 'class="dataListHeader" style="font-weight: bold;"';
    function affectedDetail($obj_db, $array_data)
    {
        $_q = $obj_db->query("SELECT action,affectedrow,title,additional_information FROM biblio_log WHERE
`date` LIKE '" . $array_data[0] . "'");
        $str = '';
        $title = '';
        if ($_q->num_rows > 0) {
            while ($_data = $_q->fetch_assoc()) {
                $title = $_data['title'];
                $str .= ' - ' . $_data['action'] . ' ' . $_data['affectedrow'] . ' : <i>' . $_data['additional_information'] .
'</i><br/>';
            }
        }
        return $title . '</br>' . $str;
    }
    // put the result into variables
    $datagrid_result = $datagrid->createDataGrid($dbs, $table_spec, 20, false);

    $_q = $dbs->query("SELECT title FROM biblio WHERE biblio_id=" . $biblioID);
    if ($_q->num_rows > 0) {
        $_d = $_q->fetch_row();
        echo '<div class="infoBox">' . __('Biblio Log') . ' : <strong>' . $_d[0] . '</strong></div>';
    }
    echo $datagrid_result;
} elseif (isset($_POST['detail']) OR (isset($_GET['action']) AND $_GET['action'] == 'detail')) {
    if ((isset($_GET['action'])) AND ($_GET['action'] == 'detail')) {
        # ADV LOG SYSTEM - STIIL EXPERIMENTAL
        $log = new AlLibrarian('1153', array("username" => $_SESSION['uname'], "uid" => $_SESSION['uid'],
"realname" => $_SESSION['realname']));
    } elseif ((isset($_GET['itemID'])) AND (isset($_GET['detail'])) AND ($_GET['detail'] == true)) {
        $log = new AlLibrarian('1155', array("username" => $_SESSION['uname'], "uid" => $_SESSION['uid'],
"realname" => $_SESSION['realname'], "biblio_id" => $_GET['itemID']));
    }
    if (!($can_read AND $can_write)) {
        die('<div class="errorBox">' . __('You are not authorized to view this section') . '</div>');
    }
    /* RECORD FORM */
    // try query
    $itemID = (integer)isset($_POST['itemID']) ? $_POST['itemID'] : 0;
    $_sql_rec_q = sprintf('SELECT b.*, p.publisher_name, pl.place_name FROM biblio AS b
    LEFT JOIN mst_publisher AS p ON b.publisher_id=p.publisher_id
    LEFT JOIN mst_place AS pl ON b.publish_place_id=pl.place_id
    WHERE biblio_id=%d', $itemID);
    $rec_q = $dbs->query($_sql_rec_q);
    $rec_d = $rec_q->fetch_assoc();
    // create new instance
    $form = new simbio_form_table_AJAX('mainForm', $_SERVER['PHP_SELF'] . '?' .
$_SERVER['QUERY_STRING'], 'post');
    $form->submit_button_attr = 'name="saveData" value="' . __('Save') . '" class="s-btn btn btn-default"';
    // form table attributes
    $form->table_attr = 'id="dataList" cellpadding="0" cellspacing="0"';
    $form->table_header_attr = 'class="alterCell"';
    $form->table_content_attr = 'class="alterCell2"';
    //custom button
    if (isset($itemID)) {
        $form->addCustomBtn('history', __('Log'), $_SERVER['PHP_SELF'] .
'?action=history&ajaxLoad=true&biblioID=' . $itemID, ' class="s-btn btn btn-success"');
```

```
    }
    $visibility = 'makeVisible s-margin__bottom-1';
    // edit mode flag set
    if ($rec_q->num_rows > 0) {
        $form->edit_mode = true;
        // record ID for delete process
        if (!$in_pop_up) {
            // form record id
            $form->record_id = $itemID;
        } else {
            $form->addHidden('updateRecordID', $itemID);
            if (isset($_POST['itemCollID'])) {
                $form->addHidden('itemCollID', $_POST['itemCollID']);
            }
            $form->back_button = false;
        }
        // form record title
        $form->record_title = $rec_d['title'];
        // submit button attribute
        $form->submit_button_attr = 'name="saveData" value="' . __('Update') . '" class="s-btn btn btn-primary"';
        // element visibility class toogle
        $visibility = 'makeHidden s-margin__bottom-1';
        // custom field data query
        $_sql_rec_cust_q = sprintf('SELECT * FROM biblio_custom WHERE biblio_id=%d', $itemID);
        $rec_cust_q = $dbs->query($_sql_rec_cust_q);
        $rec_cust_d = $rec_cust_q->fetch_assoc();
    } else {
        $_SESSION['biblioToBiblio'] = array();
    }
    // include custom fields file
    if (file_exists(MDLBS . 'bibliography/custom_fields.inc.php')) {
        include MDLBS . 'bibliography/custom_fields.inc.php';
    }
    /* Form Element(s) */
    // biblio title
    $form->addTextField('textarea', 'title', __('Title') . '*', $rec_d['title'] ?? '', 'rows="1" class="form-control"',
        __('Main title of collection. Separate child title with colon and pararel title with equal (=) sign.'));
    // biblio authors
    // $str_input = '<div class="'.$visibility.'"><a class="notAJAX button" href="javascript:
openHTMLpop(\''.MWB.'bibliography/pop_author.php?biblioID='.$rec_d['biblio_id'].'\', 500, 200,
\''.__('Authors/Roles').'\')">'.__('Add Author(s)').'</a></div>';
    $str_input = '<div class="' . $visibility . '"><a class="s-btn btn btn-default notAJAX openPopUp" href="' . MWB .
'bibliography/pop_author.php?biblioID=' . ($rec_d['biblio_id'] ?? '') . '" title="' . __('Authors/Roles') . '">' . __('Add
Author(s)') . '</a></div>';
    $str_input .= '<iframe name="authorIframe" id="authorIframe" class="form-control" style="width: 100%;
height: 100px;" src="' . MWB . 'bibliography/iframe_author.php?biblioID=' . ($rec_d['biblio_id'] ?? '') .
'&block=1"></iframe>';
    $form->addAnything(__('Author(s)'), $str_input);
    // modified by hendro wicaksono
    // biblio sor statement of responsibility
    $form->addTextField('text', 'sor', __('Statement of Responsibility'), $rec_d['sor'] ?? '', 'class="form-control"
style="width: 50%;"', __('Main source of information to show who has written, composed, illustrated, or in other ways
contributed to the existence of the item.'));
    // end of modification
    // biblio edition
    $form->addTextField('text', 'edition', __('Edition'), $rec_d['edition'] ?? '', 'class="form-control" style="width:
50%;"', __('A version of publication having substantial changes or additions.'));
    // biblio specific detail info/area
    $form->addTextField('textarea', 'specDetailInfo', __('Specific Detail Info'), $rec_d['spec_detail_info'] ?? '',
'rows="2" class="form-control', __('explain more details about an item e.g. scale within a map, running time in a
movie dvd.'));
    // biblio item batch add (by.ido alit)
    $pattern_options = array(
        // default value
        array($sysconf['batch_item_code_pattern'], $sysconf['batch_item_code_pattern'])
    );
    // get pattern from database
    $pattern_q = $dbs->query('SELECT setting_value FROM setting WHERE setting_name =
\'batch_item_code_pattern\'');
    if (!$dbs->errno) {
        // empty pattern
        $pattern_options = array(array('', '-- ' . __('Choose pattern') . ' --'));
        $pattern_d = $pattern_q->fetch_row();
        $val = @unserialize($pattern_d[0]);
        if (!empty($val)) {
            foreach ($val as $v) {
                $pattern_options[] = array($v, $v);
            }
        }
    }
    // Modified by Eddy Subratha
    // To avoid a miss processing after a pattern created, I think we should hide the Item Code Manager below
    // $str_input .= '<a href="'.MWB.'master_file/item_code_pattern.php" height="420px" class="s-btn btn btn-default
notAJAX openPopUp notIframe" title="'.__('Item code pattern manager').'">'.__('View available pattern').'</a>';
    $str_input = '<div class="form-inline">';
    $str_input .= simbio_form_element::selectList('itemCodePattern', $pattern_options, '', 'class="form-control col"') .
'&nbsp;';
    $str_input .= '<input type="text" class="form-control col-4" name="totalItems" placeholder="' . __('Total item(s)')
. '" />&nbsp;';
    $str_input .= '<div class="' . $visibility . '">
  <div class="bnt btn-group"><div class="btn btn-info" data-toggle="collapse" data-target="#batchItemDetail" aria-
expanded="false" aria-controls="batchItemDetail">' . __('Options') . '</div>';
    $str_input .= '<a href="' . MWB . 'bibliography/pop_pattern.php" height="420px" class="s-btn btn btn-default
notAJAX openPopUp notIframe"  title="' . __('Add new pattern') . '">' . __('Add New Pattern') . '</a></div></div>';
    $str_input .= '<div class="collapse" id="batchItemDetail" style="padding:10px;width:100%; text-align:left
!important;">';
    // location
    $location_q = $dbs->query("SELECT location_id, location_name FROM mst_location");
    $location_options = array(array('', '-- ' . __('Location') . ' --'));
    while ($location_d = $location_q->fetch_row()) {
        $location_options[] = array($location_d[0], $location_d[1]);
    }
    $str_input .= '<div class="form-group divRow p-1"><div class="col-3">' . __('Location') . '</div>';
    $str_input .= simbio_form_element::selectList('locationID', $location_options, '', 'class="form-control col-4"') .
'</div>';
    // item site
    $str_input .= '<div class="form-group divRow p-1"><div class="col-3">' . __('Shelf Location') . '</div>';
    $str_input .= simbio_form_element::textField('text', 'itemSite', '', 'class="form-control col-4"') . '</div>';
    // collection type
    $coll_type_q = $dbs->query("SELECT coll_type_id, coll_type_name FROM mst_coll_type");
    $coll_type_options = array(array('', '--' . __('Collection Type') . '--'));
    while ($coll_type_d = $coll_type_q->fetch_row()) {
        $coll_type_options[] = array($coll_type_d[0], $coll_type_d[1]);
    }
    $str_input .= '<div class="form-group divRow p-1"><div class="col-3">' . __('Collection Type') . '</div>';
    $str_input .= simbio_form_element::selectList('collTypeID', $coll_type_options, '', 'class="form-control col-4"') .
'</div> ';
    // item status
    $item_status_q = $dbs->query("SELECT item_status_id, item_status_name FROM mst_item_status");
    $item_status_options[] = array('0', __('Available'));
    while ($item_status_d = $item_status_q->fetch_row()) {
        $item_status_options[] = array($item_status_d[0], $item_status_d[1]);
    }
    $str_input .= '<div class="form-group divRow p-1"><div class="col-3">' . __('Item Status') . '</div>';
    $str_input .= simbio_form_element::selectList('itemStatusID', $item_status_options, '', 'class="form-control col-4"')
. '</div> ';
    // item source
    $source_options[] = array('1', __('Buy'));
    $source_options[] = array('2', __('Prize/Grant'));
    $str_input .= '<div class="form-group divRow p-1"><div class="col-3">' . __('Source') . '</div>';
    $str_input .= simbio_form_element::selectList('source', $source_options, '', 'class="form-control col-4"') . '</div> ';
    //order date
    $str_input .= '<div class="form-group divRow p-1"><div class="col-3">' . __('Order Date') . '</div>';
    $str_input .= simbio_form_element::dateField('orDate', date('Y-m-d'), 'class="form-control"') . '</div>';
    //receiving date
    $str_input .= '<div class="form-group divRow p-1"><div class="col-3">' . __('Receiving Date') . '</div>';
    $str_input .= simbio_form_element::dateField('recvDate', date('Y-m-d'), ' class="form-control col-12"') . '</div>';
    // order number
    $str_input .= '<div class="form-group divRow p-1"><div class="col-3">' . __('Order Number') . '</div>';
    $str_input .= simbio_form_element::textField('text', 'ordNo', '', 'class="form-control"') . '</div>';
    // invoice
    $str_input .= '<div class="form-group divRow p-1"><div class="col-3">' . __('Invoice') . '</div>';
    $str_input .= simbio_form_element::textField('text', 'invoice', '', 'class="form-control col-4"') . '</div>';
    // invoice date
    $str_input .= '<div class="form-group divRow p-1"><div class="col-3">' . __('Invoice Date') . '</div>';
    $str_input .= simbio_form_element::dateField('invcDate', date('Y-m-d'), ' class="form-control col-12"') . '</div>';
    // supplier
    $supplier_q = $dbs->query("SELECT supplier_id, supplier_name FROM mst_supplier");
    $supplier_options[] = array('0', __('Not Applicable'));
    while ($supplier_d = $supplier_q->fetch_row()) {
       $supplier_options[] = array($supplier_d[0], $supplier_d[1]);
    }
    $str_input .= '<div class="form-group divRow p-1"><div class="col-3">' . __('Supplier') . '</div>';
    $str_input .= simbio_form_element::selectList('supplierID', $supplier_options, '', 'class="form-control col-4"') .
'</div> ';
    //price
    $str_input .= '<div class="form-group divRow p-1"><div class="col-3">' . __('Price') . '</div>';
    $str_input .= simbio_form_element::textField('text', 'price', '0', 'class="form-control col-3"');
    $str_input .= simbio_form_element::selectList('priceCurrency', $sysconf['currencies'], '', 'class="form-control col-
2"') . '</div> ';
    $str_input .= '</div>';
    $form->addAnything(__('Item(s) code batch generator'), $str_input);
    // biblio item add
    if (!$in_pop_up AND $form->edit_mode) {
       // $str_input = '<div class="makeHidden s-margin__bottom-1"><a class="notAJAX button" href="javascript:
openHTMLpop(\''.MWB.'bibliography/pop_item.php?inPopUp=true&action=detail&biblioID='.$rec_d['biblio_id'].'\',
650, 400, \''.__('Items/Copies').'\')">'.__('Add New Items').'</a></div>';
       $str_input = '<div class="makeHidden s-margin__bottom-1"><a class="s-btn btn btn-default notAJAX
openPopUp" href="' . MWB . 'bibliography/pop_item.php?inPopUp=true&action=detail&biblioID=' .
$rec_d['biblio_id'] . '" title="' . __('Items/Copies') . '" width="780" height="500">' . __('Add New Items') .
'</a></div>';
       $str_input .= '<iframe name="itemIframe" id="itemIframe" class="form-control" style="width: 100%; height:
100px;" src="' . MWB . 'bibliography/iframe_item_list.php?biblioID=' . $rec_d['biblio_id'] . '&block=1"></iframe>' .
"\n";
       $form->addAnything(__('Item(s) Data'), $str_input);
    }
    // biblio gmd
    // get gmd data related to this record from database
    $gmd_q = $dbs->query('SELECT gmd_id, gmd_name FROM mst_gmd');
    $gmd_options = array();
    while ($gmd_d = $gmd_q->fetch_row()) {
       $gmd_options[] = array($gmd_d[0], $gmd_d[1]);
    }
    $form->addSelectList('gmdID', __('GMD'), $gmd_options, $rec_d['gmd_id'] ?? '', 'class="select2"', __('General
material designation. The physical form of publication.'));
    // biblio RDA content, media, carrier type
    foreach ($rda_cmc as $cmc => $cmc_name) {
       $cmc_options = array();
       $cmc_q = $dbs->query('SELECT id, ' . $cmc . '_type FROM mst_' . $cmc . '_type');
       $cmc_options = array();
       $cmc_options[] = array(0, __('Not set'));
       while (isset($cmc_q->num_rows) && $cmc_q->num_rows > 0 && $cmc_d = $cmc_q->fetch_row()) {
           $cmc_options[] = array($cmc_d[0], $cmc_d[1]);
       }
       if (isset($rec_d[$cmc . '_type_id'])) {
           $form->addSelectList($cmc . 'TypeID', __($cmc_name), $cmc_options, $rec_d[$cmc . '_type_id'] ?? '',
'class="select2"', __('RDA ' . $cmc_name . ' designation.'));
       } else {
           $form->addSelectList($cmc . 'TypeID', __($cmc_name), $cmc_options, '', 'class="select2"', __('RDA ' .
$cmc_name . ' designation.'));
       }
    }
    // biblio publish frequencies
    // get frequency data related to this record from database
    $freq_q = $dbs->query('SELECT frequency_id, frequency FROM mst_frequency');
    $freq_options[] = array('0', __('Not Applicable'));
    while ($freq_d = $freq_q->fetch_row()) {
       $freq_options[] = array($freq_d[0], $freq_d[1]);
    }
    $str_input = '<div class="form-inline">';
    $str_input .= simbio_form_element::selectList('frequencyID', $freq_options, $rec_d['frequency_id'] ?? '',
'class="select2 col-3"');
    $str_input .= '<div class="col">' . __('Use this for Serial publication') . '</div>';
    $str_input .= '</div>';
    $form->addAnything(__('Frequency'), $str_input);
    // biblio ISBN/ISSN
    $form->addTextField('text', 'isbn_issn', __('ISBN/ISSN'), $rec_d['isbn_issn'] ?? '', 'class="form-control"
style="width: 40%;"', __('Unique publishing number for each title of publication.'));
    // biblio publisher
    $publ_options[] = array('NONE', '');
    if (isset($rec_d['publisher_id'])) {
       $publ_q = $dbs->query(sprintf('SELECT publisher_id, publisher_name FROM mst_publisher WHERE
publisher_id=%d', $rec_d['publisher_id'] ?? ''));
       while ($publ_d = $publ_q->fetch_row()) {
           $publ_options[] = array($publ_d[0], $publ_d[1]);
       }
    }
    $form->addSelectList('publisherID', __('Publisher'), $publ_options, $rec_d['publisher_id'] ?? '', 'class="select2"
data-src="' . SWB . 'admin/AJAX_lookup_handler.php?format=json&allowNew=true" data-src-
table="mst_publisher" data-src-cols="publisher_id:publisher_name"');
    // biblio publish year
    $form->addTextField('text', 'year', __('Publishing Year'), $rec_d['publish_year'] ?? '', 'class="form-control"
style="width: 40%;"', __('Year of publication'));
    // biblio publish place
    $plc_options[] = array('NONE', '');
    if (isset($rec_d['publish_place_id'])) {
       $plc_q = $dbs->query(sprintf('SELECT place_id, place_name FROM mst_place WHERE place_id=%d',
$rec_d['publish_place_id'] ?? ''));
       while ($plc_d = $plc_q->fetch_row()) {
           $plc_options[] = array($plc_d[0], $plc_d[1]);
       }
    }
    $form->addSelectList('placeID', __('Publishing Place'), $plc_options, $rec_d['publish_place_id'] ?? '',
'class="select2" data-src="' . SWB . 'admin/AJAX_lookup_handler.php?format=json&allowNew=true" data-src-
table="mst_place" data-src-cols="place_id:place_name"');
    // biblio collation
    $form->addTextField('text', 'collation', __('Collation'), $rec_d['collation'] ?? '', 'class="form-control" style="width:
40%;"', __('Physical description of a publication e.g. publication length, width, page numbers, etc.'));
```

```
    // biblio series title
    $form->addTextField('textarea', 'seriesTitle', __('Series Title'), $rec_d['series_title'] ?? '', 'rows="1" class="form-control');
    // biblio classification
    $cls_options[] = array('NONE', '');
    if (isset($rec_d['classification'])) {
        $cls_options[] = array($rec_d['classification'], $rec_d['classification']);
    }
    $form->addSelectList('class', __('Classification'), $cls_options, $rec_d['classification'] ?? '', 'class="select2" data-src="' . SWB . 'admin/AJAX_lookup_handler.php?format=json&allowNew=true" data-src-table="mst_topic" data-src-cols="classification:classification:topic"');
    // biblio call_number
    $form->addTextField('text', 'callNumber', __('Call Number'), $rec_d['call_number'] ?? '', 'class="form-control" style="width: 40%;"', __('Sets of ID that put in the book spine.'));
    // biblio topics
    // $str_input = '<div class="'.$visibility.'"><a class="notAJAX button" href="javascript: openHTMLpop(\''.MWB.'bibliography/pop_topic.php?biblioID='.$rec_d['biblio_id'].'\', 500, 200, \''.__('Subjects/Topics').'\')">'.__('Add Subject(s)').'</a></div>';
    $str_input = '<div class="' . $visibility . ' s-margin__bottom-1"><a class="s-btn btn btn-default notAJAX openPopUp" href="' . MWB . 'bibliography/pop_topic.php?biblioID=' . ($rec_d['biblio_id'] ?? '') . '" title="' . __('Subjects/Topics') . '">' . __('Add Subject(s)') . '</a></div>';
    $str_input .= '<iframe name="topicIframe" id="topicIframe" class="form-control" style="width: 100%; height: 100px;" src="' . MWB . 'bibliography/iframe_topic.php?biblioID=' . ($rec_d['biblio_id'] ?? '') . '&block=1"></iframe>';
    $form->addAnything(__('Subject(s)'), $str_input);
    // biblio language
    // get language data related to this record from database
    $lang_q = $dbs->query("SELECT language_id, language_name FROM mst_language");
    $lang_options = array();
    while ($lang_d = $lang_q->fetch_row()) {
        $lang_options[] = array($lang_d[0], $lang_d[1]);
    }
    $form->addSelectList('languageID', __('Language'), $lang_options, $rec_d['language_id'] ?? '', 'class="select2"', __('Language use by publication.'));
    // biblio note
    $form->addTextField('textarea', 'notes', __('Abstract/Notes'), $rec_d['notes'] ?? '', 'class="form-control" style="width: 100%;" rows="3"', __('Insert here any abstract or notes from the publication.'));
    // biblio cover image
    $str_input = '<div class="row">';
    $str_input .= '<div class="col-2">';
    $str_input .= '<div id="imageFilename" class="s-margin__bottom-1">';
    $upper_dir = '';
    if ($in_pop_up) {
        $upper_dir = '../../';
    }
    if (isset($rec_d['image']) && file_exists('../../../images/docs/' . $rec_d['image'])) {
        $str_input .= '<a href="' . SWB . 'images/docs/' . ($rec_d['image'] ?? '') . '" class="openPopUp notAJAX" title="' . __('Click to enlarge preview') . '">';
        $str_input .= '<img src="' . $upper_dir . '../images/docs/' . urlencode($rec_d['image'] ?? '') . '" class="img-fluid rounded" alt="Image cover">';
        $str_input .= '</a>';
        $str_input .= '<a href="' . MWB . 'bibliography/index.php" postdata="removeImage=true&bimg=' . $itemID . '&img=' . ($rec_d['image'] ?? '') . '" loadcontainer="imageFilename" class="s-margin__bottom-1 mt-1 s-btn btn btn-danger btn-block makeHidden removeImage">' . __('Remove Image') . '</a>';
    } else {
        $str_input .= '<img src="' . $upper_dir . '../lib/minigalnano/createthumb.php?filename=images/default/image.png&width=130" class="img-fluid rounded" alt="Image cover">';
    }
    $str_input .= '</div>';
    $str_input .= '</div>';
    $str_input .= '<div class="custom-file col-7">';
    $str_input .= simbio_form_element::textField('file', 'image', '', 'class="custom-file-input" id="customFile"');
    $str_input .= '<label class="custom-file-label" for="customFile">' . __('Choose file') . '</label>';
    $str_input .= '<div style="padding: 10px;margin-left: -25px;">';
    $str_input .= '<div>' . __('Or download from url') . '</div>';
    $str_input .= '<div class="form-inline">
              <input type="text" id="getUrl" class="form-control" style="width:190px" placeholder="Paste url address
here">
              <div class="input-group-append">
              <button class="btn btn-default" type="button" id="getImage">' . __('Download') . ' <i class="fa fa-spin fa-
cog hidden" id="imgLoader"></i></button>
              <button class="btn btn-default openPopUp notAJAX ml-2" type="button" id="showEngine"
onclick="toggle_search($(\'#title\').val());">' . __('Trying search in DuckduckGo') . '</button>
              </div>
              </div>';
    $str_input .= '</div>';
    $str_input .= '</div>';
    $str_input .= ' <div class="mt-2 ml-2">Maximum ' . $sysconf['max_image_upload'] . ' KB</div>';
    $str_input .= '</div>';
    $str_input .= '<textarea id="base64picstring" name="base64picstring" style="display: none;"></textarea>';
    $str_input .= '</div></div></div>';
    //for scanner
    if ($sysconf['scanner'] !== false) {
        $str_input .= '<p>' . __('or scan a cover') . '</p>';
        // $str_input .= '<textarea id="base64picstring" name="base64picstring" style="display: none;"></textarea>';
        if ($sysconf['scanner'] == 'html5') {
            $str_input .= '<input type="button" value="' . __('Show scan dialog') . '" class="button btn btn-default
openPopUp" onclick="toggle_dialog();" />';
            $str_input .= '<input type="button" value="' . __('Reset') . '" class="button btn btn-danger openPopUp ml-1"
onclick="scan_reset();" />';
            $str_input .= '<div id="scan_overlay" style="display: none; position: absolute; left: 0; top: 0; width: 100%;
height: 100%; z-index: 1000; background: rgba(192, 194, 201, 0.5);">';
            $str_input .= '<div id="scan_dialog" title="' . __('Scan a cover') . '">';
            $str_input .= '<div id="scan_options_std" style="margin: 5px;"><label>' . __('Format:') . ' <select
id="scan_type" onchange="scan_type();">';
            $str_input .= '<option value="png">PNG</option><option value="jpg">JPEG</option></select></label> ';
            $str_input .= '<input type="button" id="btn_getscan" class="button btn" onclick="scan()" value="' .
__('Scan') . '" />';
            $str_input .= '<i style="margin-left: 10px; cursor: pointer; cursor: hand;" title="' . __('Click to show or hide
options') . '" onclick="toggle_options()" class="fa fa-gear fa-2x"></i></div>';
            $str_input .= '<div id="scan_options" class="makeHidden s-margin__bottom-1" style="margin: 5px;">';
            $str_input .= '<p style="padding: 3px 0;"><label>' . __('History index:') . ' <input type="text"
id="scan_history" value="1" style="width: 60px;" /></label> <input type="button" id="btn_getrecall"
class="button btn" onclick="scan_recall" value="' . __('Recall') . '" /></p>';
            $str_input .= '<p style="padding: 3px 0;"><label>' . __('Host:') . ' <input type="text" id="scan_host"
value="localhost" /></label> | <label>Port: <input type="text" id="scan_port" patter="\d*" maxlength="6" size="6"
style="width: 60px;" value="8811" /></label> <input type="button" id="btn_getmachine" class="button btn"
onclick="scan_init()" value="' . __('Get machine') . '" /></p>';
            $str_input .= '<p style="padding: 3px 0;"><label>' . __('Scanner:') . ' <select id="scan_machine"
readonly><option>' . __('Default') . '</option></select></label></p>';
            $str_input .= '<p style="padding: 3px 0;">' . __('Resolution') . ', <label>' . __('Horizontal:') . ' <input
type="text" id="scan_res_x" value="300" pattern="\d*" maxlength="3" size="3" style="width: 60px;"
/>dpi</label> - <label>' . __('Vertical:') . ' <input type="text" id="scan_res_y" value="300" pattern="\d*"
maxlength="3" size="3" style="width: 60px;" />dpi</label></p>';
            $str_input .= '<p style="padding: 3px 0;">' . __('Capture') . ', <label>' . __('Width:') . ' <input type="text"
id="scan_capture_w" value="2550" pattern="\d*" maxlength="4" size="4" style="width: 60px;" />px</label> -
<label>' . __('Height:') . ' <input type="text" id="scan_capture_h" value="3507" pattern="\d*" maxlength="4"
size="4" style="width: 60px;" />px</label></p>';
            $str_input .= '<p>' . __('Result') . ', <label>' . __('Max Width:') . ' <input type="text" id="scan_max_w"
value="360" pattern="\d*" maxlength="3" size="3" style="width: 60px;" />px</label> - <label>' . __('Max Height:')
. ' <input type="text" id="scan_max_h" value="480" pattern="\d*" maxlength="3" size="3" style="width: 60px;"
/>px</label></p></div>';
```

```
            $str_input .= '<div id="scan_container" style="margin: 5px;"><div style="height: 550px; width: 390px;
overflow: auto; float: left;"><p>' . __('Scan result') . '</p><img id="my_imgdata" style="margin: auto;" /></div>';
            $str_input .= '<div style="padding-left: 10px; height: 550; width: 400px; overflow: auto; float: left;"><p>' .
__('Preview') . ' <input type="button" class="button btn" value="' . __('Rotate Left') . '"
onclick="scan_rotate(\'left\')" /> <input type="button" class="button btn" value="' . __('Rotate Right') . '"
onclick="scan_rotate(\'right\')" /></p><canvas id="my_selected" style="border: 1px solid #CCC; margin:
auto;"></canvas></div></div></div></div>';
        }
    }
    $form->addAnything(__('Image'), $str_input);
    // biblio file attachment
    // $str_input = '<div class="'.$visibility.'"><a class="notAJAX button" href="javascript:
openHTMLpop(\''.MWB.'bibliography/pop_attach.php?biblioID='.$rec_d['biblio_id'].'\', 600, 300, \''.__('File
Attachments').'\')">'.__('Add Attachment').'</a></div>';
    $str_input = '<div class="' . $visibility . ' s-margin__bottom-1"><a class="s-btn btn btn-default notAJAX
openPopUp" href="' . MWB . 'bibliography/pop_attach.php?biblioID=' . ($rec_d['biblio_id'] ?? '') . '" width="780"
height="500" title="' . __('File Attachments') . '">' . __('Add Attachment') . '</a></div>';
    $str_input .= '<iframe name="attachIframe" id="attachIframe" class="form-control" style="width: 100%; height:
100px;" src="' . MWB . 'bibliography/iframe_attach.php?biblioID=' . ($rec_d['biblio_id'] ?? '') .
'&block=1"></iframe>';
    $form->addAnything(__('File Attachment'), $str_input);
    // biblio relation
    $str_input = '<div class="' . $visibility . ' s-margin__bottom-1"><a class="s-btn btn btn-default notAJAX
openPopUp" href="' . MWB . 'bibliography/pop_biblio_rel.php?biblioID=' . ($rec_d['biblio_id'] ?? '') . '" title="' .
__('Biblio Relation') . '">' . __('Add Relation') . '</a></div>';
    $str_input .= '<iframe name="biblioIframe" id="biblioIframe" class="form-control" style="width: 100%; height:
100px;" src="' . MWB . 'bibliography/iframe_biblio_rel.php?biblioID=' . ($rec_d['biblio_id'] ?? '') .
'&block=1"></iframe>';
    $form->addAnything(__('Related Biblio Data'), $str_input);
    /**
     * Custom fields
     */
    if (isset($biblio_custom_fields)) {
        if (is_array($biblio_custom_fields) && $biblio_custom_fields) {
            foreach ($biblio_custom_fields as $fid => $cfield) {
                // custom field properties
                $cf_dbfield = $cfield['dbfield'];
                $cf_label = $cfield['label'];
                $cf_default = $cfield['default'];
                $cf_class = $cfield['class'] ?? '';
                $cf_note = $cfield['note'] ?? '';
                $cf_data = (isset($cfield['data']) && $cfield['data']) ? unserialize($cfield['data']) : array();
                // get data field record
                if (isset($rec_cust_d[$cf_dbfield]) && @unserialize($rec_cust_d[$cf_dbfield]) !== false) {
                    $rec_cust_d[$cf_dbfield] = unserialize($rec_cust_d[$cf_dbfield]);
                }
                // custom field processing
                if (in_array($cfield['type'], array('text', 'longtext', 'numeric'))) {
                    $cf_max = isset($cfield['max']) ? $cfield['max'] : '200';
                    $cf_width = isset($cfield['width']) ? $cfield['width'] : '50';
                    $form->addTextField(($cfield['type'] == 'longtext') ? 'textarea' : 'text', $cf_dbfield, $cf_label,
$rec_cust_d[$cf_dbfield] ?? $cf_default, ' class="form-control ' . $cf_class . '" style="width: ' . $cf_width . '%;"
maxlength="' . $cf_max . '"', $cf_note);
                } else if ($cfield['type'] == 'dropdown') {
                    $form->addSelectList($cf_dbfield, $cf_label, $cf_data, $rec_cust_d[$cf_dbfield] ?? $cf_default, '
class="form-control ' . $cf_class . '"');
                } else if ($cfield['type'] == 'checklist') {
                    $form->addCheckBox($cf_dbfield, $cf_label, $cf_data, $rec_cust_d[$cf_dbfield] ?? $cf_default, '
class="form-control ' . $cf_class . '"');
                } else if ($cfield['type'] == 'choice') {
                    $form->addRadio($cf_dbfield, $cf_label, $cf_data, $rec_cust_d[$cf_dbfield] ?? $cf_default, ' class="form-
control ' . $cf_class . '"');
```

```
            } else if ($cfield['type'] == 'date') {
              $form->addDateField($cf_dbfield, $cf_label, $rec_cust_d[$cf_dbfield] ?? NULL, ' class="form-control ' .
$cf_class . '"');
            }
            unset($cf_data);
          }
        }
    }
    // biblio hide from opac
    $hide_options[] = array('0', __('Show'));
    $hide_options[] = array('1', __('Hide'));
    $form->addRadio('opacHide', __('Hide in OPAC'), $hide_options, isset($rec_d['opac_hide']) &&
$rec_d['opac_hide'] ? '1' : '0');
    // biblio promote to front page
    $promote_options[] = array('0', __('Don\'t Promote'));
    $promote_options[] = array('1', __('Promote'));
    $form->addRadio('promote', __('Promote To Homepage'), $promote_options, isset($rec_d['promoted']) &&
$rec_d['promoted'] ? '1' : '0');
    // biblio labels
    $arr_labels = !empty($rec_d['labels']) ? unserialize($rec_d['labels']) : array();
    if ($arr_labels) {
        foreach ($arr_labels as $label) {
          $arr_labels[$label[0]] = $label[1];
        }
    }
    $str_input = '';
    // get label data from database
    // Modified by Eddy Subratha
    $label_q = $dbs->query("SELECT * FROM mst_label LIMIT 20");
    while ($label_d = $label_q->fetch_assoc()) {
        $checked = isset($arr_labels[$label_d['label_name']]) ? ' checked' : '';
        $url = isset($arr_labels[$label_d['label_name']]) ? $arr_labels[$label_d['label_name']] : '';
        $str_input .= '
       <div class="input-group s-labels__group noAutoFocus">
        <div class="input-group-prepend">
          <span class="input-group-text" style="width:150px">
            <input type="checkbox" name="labels[]" value="' . $label_d['label_name'] . '"' . $checked . ' aria-
label="Labels">&nbsp;' . $label_d['label_desc'] . '
          </span>
          <span class="input-group-text s-labels__icon">
            <img src="../lib/minigalnano/createthumb.php?filename=' . IMG . '/labels/' . urlencode($label_d['label_image'])
. '&amp;width=24" class="' . $label_d['label_name'] . '" id="' . $label_d['label_name'] . '" alt="' .
$label_d['label_name'] . '"/>
          </span>
        </div>
        <input type="text" value="' . $url . '" placeholder="Enter a website link/URL to make this label clickable"
title="Enter a website link/URL to make this label clickable" name="label_urls[' . $label_d['label_name'] . ']"
id="label_urls[' . $label_d['label_name'] . ']" class="form-control" aria-label="Url for current label" style="border-
left:none;">
        </div>';
    }
    $form->addAnything('Label', $str_input);
    // $form->addCheckBox('labels', 'Label', $label_options, explode(' ', $rec_d['labels']));
    // edit mode messagge
    if ($form->edit_mode) {
        echo '<div class="s-alert infoBox">'
          . __('You are going to edit biblio data') . ' : <b>' . $rec_d['title'] . '</b>  <br />' . __('Last Updated') . '&nbsp;' .
date('d F Y h:i:s', strtotime($rec_d['last_update'])); //mfc
        if (isset($rec_d['image'])) {
          if (file_exists(IMGBS . 'docs/' . $rec_d['image'])) {
            $upper_dir = '';
            if ($in_pop_up) {
                $upper_dir = '../../';
            }
            // Modified by Eddy Subratha
            // We removed image near the upload form
            // echo '<div id="biblioImage" class="s-biblio__cover"><img
src="'.$upper_dir.'../lib/minigalnano/createthumb.php?`filename`=../../images/docs/'.urlencode($rec_d['image']).'&wi
dth=53" /></div>';
        }
    }
    echo '</div>' . "\n";
  }
  // print out the form object
  echo $form->printOut();
  // javascript
  ?>
  <script type="text/javascript">
    $(document).ready(function () {
        // popup pattern
        $('#class').change(function () {
            $('#callNumber').val($(this).val().replace('NEW:', ''));
        });
        $('.removeImage').click(function (e) {
            if (confirm('Are you sure you want to permanently remove this image?')) {
                return true;
            } else {
                return false;
            }
        });

        $(document).on('change', '.custom-file-input', function () {
            // $('#imageFilename img').attr('src',document.getElementById("image").files[0].name);
            var input = document.querySelector("#image");
            var fReader = new FileReader();
            fReader.readAsDataURL(input.files[0]);
            fReader.onloadend = function (event) {
                var img = document.querySelector("#imageFilename img");
                img.src = event.target.result;
            }
            let fileName = $(this).val().replace(/\\/g, '/').replace(/.*\//, '');
            $(this).parent('.custom-file').find('.custom-file-label').text(fileName);
        });
        $('#getImage').click(function () {
            $.post("<?php echo MWB ?>bibliography/scrape_image.php", {imageURL: $('#getUrl').val()})
                .done(function (data) {
                    if (data.status == 'VALID') {
                        $('#base64picstring').val(data.image);
                        $('#imageFilename img').attr('src', data.message);
                    } else {
                        $('#base64picstring, #getUrl').val('');
                        parent.toastr.error("<?php echo __('Current url is not valid or your internet is down.') ?>",
"Bibliography Image", {
                            "closeButton": true,
                            "debug": false,
                            "newestOnTop": false,
                            "progressBar": false,
                            "positionClass": "toast-top-right",
                            "preventDuplicates": false,
                            "onclick": null,
                            "showDuration": 300,
                            "hideDuration": 1000,
                            "timeOut": 5000,
                            "extendedTimeOut": 1000,
                            "showEasing": "swing",
                            "hideEasing": "linear",
                            "showMethod": "fadeIn",
                            "hideMethod": "fadeOut"
                        })
                    }
                });
        });
    });
    </script>
    <?php
} else {
    # ADV LOG SYSTEM - STIIL EXPERIMENTAL
    $log = new AlLibrarian('1151', array("username" => $_SESSION['uname'], "uid" => $_SESSION['uid'],
"realname" => $_SESSION['realname']));
    require SIMBIO . 'simbio_UTILS/simbio_tokenizecql.inc.php';
    require MDLBS . 'bibliography/biblio_utils.inc.php';
    require LIB . 'biblio_list_model.inc.php';
    // number of records to show in list
    $biblio_result_num = ($sysconf['biblio_result_num'] > 100) ? 100 : $sysconf['biblio_result_num'];
    // create datagrid
    $datagrid = new simbio_datagrid();
    // index choice
    if ($sysconf['index']['type'] == 'index' || $sysconf['index']['type'] == 'sphinx') {
        if ($sysconf['index']['type'] == 'sphinx') {
        require LIB . 'sphinx/sphinxapi.php';
        require LIB . 'biblio_list_sphinx.inc.php';
        } else {
        require LIB . 'biblio_list_index.inc.php';
        }
        // table spec
        $table_spec = 'search_biblio AS `index` LEFT JOIN item ON `index`.biblio_id=item.biblio_id';
        $str_criteria = 'index.biblio_id IS NOT NULL';
        if ($can_read AND $can_write) {
        $datagrid->setSQLColumn('index.biblio_id', 'index.title AS \'' . __('Title') . '\'', 'index.labels', 'index.image',
            'index.author',
            'index.isbn_issn AS \'' . __('ISBN/ISSN') . '\'',
            'IF(COUNT(item.item_id)>0, COUNT(item.item_id), \'<strong style="color: #f00;">' . __('None') .
'</strong>\') AS \'' . __('Copies') . '\'',
            'index.last_update AS \'' . __('Last Update') . '\'');
        $datagrid->modifyColumnContent(1, 'callback{showTitleAuthors}');
        } else {
        $datagrid->setSQLColumn('index.title AS \'' . __('Title') . '\'', 'index.author', 'index.labels', 'index.image',
            'index.isbn_issn AS \'' . __('ISBN/ISSN') . '\'',
            'IF(COUNT(item.item_id)>0, COUNT(item.item_id), \'<strong style="color: #f00;">' . __('None') .
'</strong>\') AS \'' . __('Copies') . '\'',
            'index.last_update AS \'' . __('Last Update') . '\'');
        $datagrid->modifyColumnContent(1, 'callback{showTitleAuthors}');
        }
        $datagrid->invisible_fields = array(1, 2, 3);
        $datagrid->setSQLorder('index.last_update DESC');
        // set group by
        $datagrid->sql_group_by = 'index.biblio_id';
    } else {
        require LIB . 'biblio_list.inc.php';
        // table spec
        $table_spec = 'biblio LEFT JOIN item ON biblio.biblio_id=item.biblio_id';
        $str_criteria = 'biblio.biblio_id IS NOT NULL';
        if ($can_read AND $can_write) {
        $datagrid->setSQLColumn('biblio.biblio_id', 'biblio.biblio_id AS bid',
            'biblio.title AS \'' . __('Title') . '\'',
            'biblio.isbn_issn AS \'' . __('ISBN/ISSN') . '\'',
        'IF(COUNT(item.item_id)>0, COUNT(item.item_id), \'<strong style="color: #f00;">' . __('None') .
'</strong>\') AS \'' . __('Copies') . '\'',
        'biblio.last_update AS \'' . __('Last Update') . '\'');
        $datagrid->modifyColumnContent(2, 'callback{showTitleAuthors}');
    } else {
        $datagrid->setSQLColumn('biblio.biblio_id AS bid', 'biblio.title AS \'' . __('Title') . '\'',
        'biblio.isbn_issn AS \'' . __('ISBN/ISSN') . '\'',
        'IF(COUNT(item.item_id)>0, COUNT(item.item_id), \'<strong style="color: #f00;">' . __('None') .
'</strong>\') AS \'' . __('Copies') . '\'',
        'biblio.last_update AS \'' . __('Last Update') . '\'');
        // modify column value
        $datagrid->modifyColumnContent(1, 'callback{showTitleAuthors}');
    }
    $datagrid->invisible_fields = array(0);
    $datagrid->setSQLorder('biblio.last_update DESC');
    // set group by
    $datagrid->sql_group_by = 'biblio.biblio_id';
  }

  $stopwords = "@\sAnd\s|\sOr\s|\sNot\s|\sThe\s|\sDan\s|\sAtau\s|\sAn\s|\sA\s@i";
  // is there any search
  if (isset($_GET['keywords']) AND $_GET['keywords']) {
    $keywords = $dbs->escape_string(trim($_GET['keywords']));
    $keywords = preg_replace($stopwords, ' ', $keywords);
    $searchable_fields = array('title', 'author', 'subject', 'isbn', 'publisher');
    if ($_GET['field'] != '0' AND in_array($_GET['field'], $searchable_fields)) {
      $field = $_GET['field'];
      $search_str = $field . '=' . $keywords;
    } else {
      $search_str = '';
      foreach ($searchable_fields as $search_field) {
        $search_str .= $search_field . '=' . $keywords . ' OR ';
      }
      $search_str = substr_replace($search_str, '', -4);
    }
    $biblio_list = new biblio_list($dbs, $biblio_result_num);
    $criteria = $biblio_list->setSQLcriteria($search_str);
    $str_criteria .= ' AND (' . $criteria['sql_criteria'] . ')';
  }
  if (isset($_GET['opac_hide']) && $_GET['opac_hide'] != '') {
    $opac_hide = $dbs->escape_string($_GET['opac_hide']);
    $str_criteria .= ' AND opac_hide =' . $opac_hide;
  }
  if (isset($_GET['promoted']) && $_GET['promoted'] != '') {
    $promoted = $dbs->escape_string($_GET['promoted']);
    $str_criteria .= ' AND promoted =' . $promoted;
  }
  //debug
  //echo $str_criteria;
  $datagrid->setSQLcriteria($str_criteria);
  // set table and table header attributes
  $datagrid->table_attr = 'id="dataList" class="s-table table"';
  $datagrid->table_header_attr = 'class="dataListHeader" style="font-weight: bold;"';
  // set delete proccess URL
  $datagrid->chbox_form_URL = $_SERVER['PHP_SELF'];
  $datagrid->debug = true;
  // put the result into variables
  $datagrid_result = $datagrid->createDataGrid($dbs, $table_spec, $biblio_result_num, ($can_read AND
$can_write));
  if (isset($_GET['keywords']) AND $_GET['keywords']) {
    $msg = str_replace('{result->num_rows}', $datagrid->num_rows, __('Found <strong>{result-
>num_rows}</strong> from your keywords')); //mfc
```

```
        echo '<div class="infoBox">' . $msg . ' : "' . htmlentities($_GET['keywords']) . '"<div>' . __('Query took') . ' <b>'
. $datagrid->query_time . '</b> ' . __('second(s) to complete') . '</div></div>'; //mfc
    }
    echo $datagrid_result;
}
/* main content end */
```

Berikut adalah potongan *source c*ode tampilan halaman Keanggotaan perpustakaan.

```
<?php
// key to authenticate
define('INDEX_AUTH', '1');
// key to get full database access
define('DB_ACCESS', 'fa');
if (!defined('SB')) {
  // main system configuration
  require '../../../sysconfig.inc.php';
  // start the session
  require SB.'admin/default/session.inc.php';
}
// IP based access limitation
require LIB.'ip_based_access.inc.php';
do_checkIP('smc');
do_checkIP('smc-membership');
require SB.'admin/default/session_check.inc.php';
require SIMBIO.'simbio_GUI/table/simbio_table.inc.php';
require SIMBIO.'simbio_GUI/form_maker/simbio_form_table_AJAX.inc.php';
require SIMBIO.'simbio_GUI/paging/simbio_paging.inc.php';
require SIMBIO.'simbio_DB/datagrid/simbio_dbgrid.inc.php';
require SIMBIO.'simbio_DB/simbio_dbop.inc.php';
require SIMBIO.'simbio_UTILS/simbio_date.inc.php';
require SIMBIO.'simbio_FILE/simbio_file_upload.inc.php';
// privileges checking
$can_read = utility::havePrivilege('membership', 'r');
$can_write = utility::havePrivilege('membership', 'w');
if (!$can_read) {
    die('<div class="errorBox">You dont have enough privileges to view this section</div>');
}
// execute registered hook
\SLiMS\Plugins::getInstance()->execute('membership_init');
/* Just In Case for PHP < 5.4 */
/* Taken From imageman (http://www.php.net/manual/en/function.getimagesizefromstring.php#113976) */
/* Make sure to set allow_url_fopen = on inside your php.ini */
if (version_compare(phpversion(), '5.4', '<'))
{
    function getimagesizefromstring($string_data)
    {
        $uri = 'data://application/octet-stream;base64,'  . base64_encode($string_data);
        return getimagesize($uri);
    }
}
/* REMOVE IMAGE */
if (isset($_POST['removeImage']) && isset($_POST['mimg']) && isset($_POST['img'])) {
  // validate post image
  $member_id = utility::filterData('mimg', 'post', true, true, true);
  $image_name = utility::filterData('img', 'post', true, true, true);

  $query_image = $dbs->query("SELECT member_id FROM member WHERE member_id='{$member_id}' AND
member_image='{$image_name}'");
  if (!empty($query_image->num_rows)) {
    $_delete = $dbs->query(sprintf('UPDATE member SET member_image=NULL WHERE member_id=%d',
$member_id));
    if ($_delete) {
      $postImage = stripslashes($_POST['img']);
      $postImage = str_replace('/', '', $postImage);
      @unlink(sprintf(IMGBS.'persons/%s', $postImage));
      exit('<script type="text/javascript">alert(\''.str_replace('{imageFilename}', $postImage, __('{imageFilename}
successfully removed!')).'\'); $(\'#memberImage, #imageFilename\').remove();</script>');
    }
  }
  exit();
}
/* member update process */
if (isset($_POST['saveData']) AND $can_read AND $can_write) {
    // check form validity
    $memberID = trim($_POST['memberID']);
    $memberName = trim($_POST['memberName']);
    $birthDate = trim($_POST['birthDate']);
    $mpasswd1 = trim($_POST['memberPasswd']);
    $mpasswd2 = trim($_POST['memberPasswd2']);
    if (empty($memberID) OR empty($memberName) OR empty($birthDate)) {
        utility::jsAlert(__('Member ID, Name and Birthday cannot be empty')); //mfc
        exit();
    } else if (($mpasswd1 OR $mpasswd2) AND ($mpasswd1 !== $mpasswd2)) {
        utility::jsAlert(__('Password confirmation does not match. See if your Caps Lock key is on!'));
        exit();
    } else {
        // include custom fields file
        if (file_exists(MDLBS.'membership/member_custom_fields.inc.php')) {
            include MDLBS.'membership/member_custom_fields.inc.php';
        }
        /**
         * Custom fields
         */
        if (isset($member_custom_fields)) {
          if (is_array($member_custom_fields) && $member_custom_fields) {
            foreach ($member_custom_fields as $fid => $cfield) {
              // custom field data
              $cf_dbfield = $cfield['dbfield'];
              if (isset($_POST[$cf_dbfield])) {
                if(is_array($_POST[$cf_dbfield])){
                  foreach ($_POST[$cf_dbfield] as $value) {
                    $arr[$value] = $value;
                  }
                  $custom_data[$cf_dbfield] = serialize($arr);
                }
                else{
                  $cf_val = $dbs->escape_string(strip_tags(trim($_POST[$cf_dbfield]),
$sysconf['content']['allowable_tags']));
                  if($cfield['type'] == 'numeric' && (!is_numeric($cf_val) && $cf_val!='')){
                    utility::jsToastr(__('Membership'), sprintf(__('Field %s only number for allowed'),$cfield['label']),
'error');
                    exit();
                  }
                  elseif ($cfield['type'] == 'date' && $cf_val == '') {
                    utility::jsToastr(__('Membership'), sprintf(__('Field %s is date format, empty not
allowed'),$cfield['label']), 'error');
                    exit();
                  }
                  $custom_data[$cf_dbfield] = $cf_val;
                }
              }else{
```

```
            $custom_data[$cf_dbfield] = serialize(array());
          }
        }
      }
    }
    $data['member_id'] = $dbs->escape_string($memberID);
    $data['member_name'] = $dbs->escape_string($memberName);
    $data['member_type_id'] = (integer)$_POST['memberTypeID'];
    $data['inst_name'] = trim($dbs->escape_string(strip_tags($_POST['instName'])));
    $data['gender'] = trim($dbs->escape_string(strip_tags($_POST['gender'])));
    $data['birth_date'] = trim($dbs->escape_string(strip_tags($_POST['birthDate'])));
    $data['birth_date'] = $data['birth_date'] == '' ? null : $data['birth_date'];
    $data['register_date'] = trim($dbs->escape_string(strip_tags($_POST['regDate'])));
    // member since date
    $member_since = trim($dbs->escape_string(strip_tags($_POST['sinceDate'])));
    if (isset($_POST['updateRecordID'])) {
        $data['member_since_date'] = $member_since;
    } else {
        if ($member_since) {
            $data['member_since_date'] = $member_since;
        } else {
            $data['member_since_date'] = $data['register_date'];
        }
    }
    $data['expire_date'] = trim($dbs->escape_string(strip_tags($_POST['expDate'])));
    // extending membership
    if (isset($_POST['extend']) AND !empty($_POST['extend'])) {
        // get membership periode from database
        $mtype_query = $dbs->query("SELECT member_periode FROM mst_member_type WHERE
member_type_id=".$data['member_type_id']);
        $mtype_data = $mtype_query->fetch_row();
        $data['register_date'] = date('Y-m-d');
        $data['expire_date'] = simbio_date::getNextDate($mtype_data[0], $data['register_date']);
    }
    $data['pin'] = trim($dbs->escape_string(strip_tags($_POST['memberPIN'])));
    $data['member_address'] = trim($dbs->escape_string(strip_tags($_POST['memberAddress'])));
    $data['member_mail_address'] = trim($dbs->escape_string(strip_tags($_POST['memberMailAddress'])));
    $data['member_phone'] = trim($dbs->escape_string(strip_tags($_POST['memberPhone'])));
    $data['member_fax'] = trim($dbs->escape_string(strip_tags($_POST['memberFax'])));
    $data['postal_code'] = trim($dbs->escape_string(strip_tags($_POST['memberPostal'])));
    $data['member_notes'] = trim($dbs->escape_string(strip_tags($_POST['memberNotes'])));
    $data['member_email'] = trim($dbs->escape_string(strip_tags($_POST['memberEmail'])));
    $data['is_pending'] = isset($_POST['isPending'])? intval($_POST['isPending']) : '0';
    $data['input_date'] = date('Y-m-d');
    $data['last_update'] = date('Y-m-d');
    if (!empty($_FILES['image']) AND $_FILES['image']['size']) {
      // create upload object
      $upload = new simbio_file_upload();
      $upload->setAllowableFormat($sysconf['allowed_images']);
      $upload->setMaxSize($sysconf['max_image_upload']*1024); // approx. 100 kb
      $upload->setUploadDir(IMGBS.'persons');
      // give new name for upload file
      $new_filename = 'member_'.$data['member_id'];
      $upload_status = $upload->doUpload('image', $new_filename);
      if ($upload_status == UPLOAD_SUCCESS) {
        $data['member_image'] = $dbs->escape_string($upload->new_filename);
      }
    } else if (!empty($_POST['base64picstring'])) {
                                            list($filedata, $filedom) = explode('#image/type#', $_POST['base64picstring']);
      $filedata = base64_decode($filedata);
      $fileinfo = getimagesizefromstring($filedata);
      $valid = strlen($filedata)/1024 < $sysconf['max_image_upload'];
```

```
        $valid = (!$fileinfo || $valid === false) ? false : in_array($fileinfo['mime'],
$sysconf['allowed_images_mimetype']);
                                $new_filename = 'member_'.$data['member_id'].'.'.strtolower($filedom);

                                if ($valid AND file_put_contents(IMGBS.'persons/'.$new_filename, $filedata)) {
                                        $data['member_image'] = $dbs->escape_string($new_filename);
                                        if (!defined('UPLOAD_SUCCESS')) define('UPLOAD_SUCCESS', 1);
                                        $upload_status = UPLOAD_SUCCESS;
                                }
                        }
      // password confirmation
      if (($mpasswd1 AND $mpasswd2) AND ($mpasswd1 === $mpasswd2)) {
        // $data['mpasswd'] = 'literal{MD5(\''.$mpasswd2.'\')}';
        $data['mpasswd'] = password_hash($mpasswd2, PASSWORD_BCRYPT);
      }
      // create sql op object
      $sql_op = new simbio_dbop($dbs);
      if (isset($_POST['updateRecordID'])) {
          /* UPDATE RECORD MODE */
          // remove input date
          unset($data['input_date']);
          // filter update record ID
          $updateRecordID = $dbs->escape_string(trim($_POST['updateRecordID']));
          $old_member_ID = $updateRecordID;
          // update the data
          $update = $sql_op->update('member', $data, "member_id='$updateRecordID'");
          if ($update) {
              // execute registered hook
              \SLiMS\Plugins::getInstance()->execute('membership_after_update', ['data' => api::member_load($dbs,
$updateRecordID)]);
              // update custom data
              if (isset($custom_data)) {
                // check if custom data for this record exists
                $_sql_check_custom_q = sprintf('SELECT member_id FROM member_custom WHERE member_id=%d',
$updateRecordID);
                $check_custom_q = $dbs->query($_sql_check_custom_q);
                if ($check_custom_q->num_rows) {
                  @$sql_op->update('member_custom', $custom_data, 'member_id=\''.$updateRecordID.'\'');
                } else {
                  $custom_data['member_id'] = $updateRecordID;
                  @$sql_op->insert('member_custom', $custom_data);
                }
              }
              // update other tables contain this member ID
              @$dbs->query('UPDATE loan SET member_id=\''.$data['member_id'].'\' WHERE
member_id=\''.$old_member_ID.'\'');
              @$dbs->query('UPDATE fines SET member_id=\''.$data['member_id'].'\' WHERE
member_id=\''.$old_member_ID.'\'');
              utility::jsAlert(__('Member Data Successfully Updated'));
              // upload status alert
              if (isset($upload_status)) {
                  if ($upload_status == UPLOAD_SUCCESS) {
                      // write log
                      utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'membership', $_SESSION['realname'].' upload
image file '.$upload->new_filename, 'Photo', 'Update');
                      utility::jsAlert(__('Image Uploaded Successfully'));
                  } else {
                      // write log
                      utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'membership', 'ERROR : '.$_SESSION['realname'].'
FAILED TO upload image file '.$upload->new_filename.', with error ('.$upload->error.')', 'Photo', 'Fail');
                      utility::jsAlert(__('Image FAILED to upload'));
                  }
```

```
            }
            // write log
            utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'membership', $_SESSION['realname'].' update member
data ('.$memberName.') with ID ('.$memberID.')', 'Update', 'OK');
            if ($sysconf['webcam'] == 'html5') {
              echo '<script
type="text/javascript">parent.$(\'#mainContent\').simbioAJAX(\''.MWB.'membership/index.php\');</script>';
            } else {
              echo '<script
type="text/javascript">parent.$(\'#mainContent\').simbioAJAX(\''.MWB.'membership/index.php\');</script>';
            }
          } else { utility::jsAlert(__('Member Data FAILED to Save/Update. Please Contact System
Administrator')."\nDEBUG : ".$sql_op->error); }
          exit();
      } else {
          /* INSERT RECORD MODE */
          if (!$mpasswd1 AND !$mpasswd2) { $data['mpasswd'] = 'literal{NULL}'; }
          // insert the data
          $insert = $sql_op->insert('member', $data);
          if ($insert) {
            // insert custom data
            if ($custom_data) {
              $custom_data['member_id'] = $data['member_id'];
              @$sql_op->insert('member_custom', $custom_data);
            }
            \SLiMS\Plugins::getInstance()->execute('membership_after_save', ['data' => api::member_load($dbs,
$data['member_id'])]);
            utility::jsAlert(__('New Member Data Successfully Saved'));
            // upload status alert
            if (isset($upload_status)) {
                if ($upload_status == UPLOAD_SUCCESS) {
                    // write log
                    utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'membership', $_SESSION['realname'].' upload
image file '.$upload->new_filename, 'Photo', 'Add');
                    utility::jsAlert(__('Image Uploaded Successfully'));
                } else {
                    // write log
                    utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'membership', 'ERROR : '.$_SESSION['realname'].'
FAILED TO upload image file '.$upload->new_filename.', with error ('.$upload->error.')', 'Photo', 'Fail');
                    utility::jsAlert(__('Image FAILED to upload'));
                }
            }
            // write log
            utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'membership', $_SESSION['realname'].' add new member
('.$memberName.') with ID ('.$memberID.')', 'Add', 'OK');
            echo '<script
type="text/javascript">parent.$(\'#mainContent\').simbioAJAX(\''.$_SERVER['PHP_SELF'].'\');</script>';
          } else { utility::jsAlert(__('Member Data FAILED to Save/Update. Please Contact System
Administrator')."\nDEBUG : ".$sql_op->error); }
          exit();
      }
   }
   exit();
} else if (isset($_POST['batchExtend']) && $can_read && $can_write) {
   /* BATCH extend membership proccessing */
   $curr_date = date('Y-m-d');
   $num_extended = 0;
   foreach ($_POST['itemID'] as $itemID) {
      $memberID = $dbs->escape_string(trim($itemID));
      // get membership periode from database
      $mtype_q = $dbs->query('SELECT member_periode, m.member_name FROM member AS m
          LEFT JOIN mst_member_type AS mt ON m.member_type_id=mt.member_type_id
          WHERE m.member_id=\''.$memberID.'\'');
        $mtype_d = $mtype_q->fetch_row();
        $expire_date = simbio_date::getNextDate($mtype_d[0], $curr_date);
        @$dbs->query('UPDATE member SET register_date=\''.date("Y-m-d").'\',  expire_date=\''.$expire_date.'\',
last_update=\''.date("y-m-d").'\' WHERE member_id=\''.$memberID.'\'');
        // write log
        utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'membership', $_SESSION['realname'].' extends membership
for member ('.$mtype_d[1].') with ID ('.$memberID.')', 'Extend', 'OK');
        $num_extended++;
    }
    header('Location: '.MWB.'membership/index.php?expire=true&numExtended='.$num_extended);
    exit();
} else if (isset($_POST['itemID']) AND !empty($_POST['itemID']) AND isset($_POST['itemAction'])) {
    if (!($can_read AND $can_write)) {
        die();
    }
    /* DATA DELETION PROCESS */
    $sql_op = new simbio_dbop($dbs);
    $failed_array = array();
    $error_num = 0;
    $still_have_loan = array();
    if (!is_array($_POST['itemID'])) {
        // make an array
        $_POST['itemID'] = array($dbs->escape_string(trim($_POST['itemID'])));
    }
    // loop array
    foreach ($_POST['itemID'] as $itemID) {
        $itemID = $dbs->escape_string(trim($itemID));
        // check if the member still have loan
        $loan_q = $dbs->query('SELECT DISTINCT m.member_id, m.member_name, COUNT(l.loan_id) FROM
member AS m
            LEFT JOIN loan AS l ON (m.member_id=l.member_id AND l.is_lent=1 AND l.is_return=0)
            WHERE m.member_id=\''.$itemID.'\' GROUP BY m.member_id');
        $loan_d = $loan_q->fetch_row();
        if ($loan_d[2] < 1) {
            if (!$sql_op->delete('member', "member_id='$itemID'")) {
                $error_num++;
            } else {
                // write log
                utility::writeLogs($dbs, 'staff', $_SESSION['uid'], 'membership', $_SESSION['realname'].' DELETE
member data ('.$loan_d[1].') with ID ('.$loan_d[0].')', 'Delete', 'OK');
            }
        } else {
            $still_have_loan[] = $loan_d[0].' - '.$loan_d[1];
            $error_num++;
        }
    }
    if ($still_have_loan) {
        $members = '';
        foreach ($still_have_loan as $mbr) {
            $members .= $mbr."\n";
        }
        utility::jsAlert(__('Below member data can\'t be deleted because still have unreturned item(s)').' : '."\n".$mbr);
        exit();
    }
    // error alerting
    if ($error_num == 0) {
        utility::jsAlert(__('All Data Successfully Deleted'));
        echo '<script
type="text/javascript">parent.$(\'#mainContent\').simbioAJAX(\''.$_SERVER['PHP_SELF'].'?'.$_POST['lastQueryS
tr'].'\');</script>';
    } else {
        utility::jsAlert(__('Some or All Data NOT deleted successfully!\nPlease contact system administrator'));
        echo '<script
type="text/javascript">parent.$(\'#mainContent\').simbioAJAX(\''.$_SERVER['PHP_SELF'].'?'.$_POST['lastQuerySt
r'].'\');</script>';
    }
    exit();
}
/* RECORD OPERATION END */

/* search form */
$page_title = __('Membership');
if(isset($_GET['expire'])) {
    $page_title = __('Expired Member List');
}
?>
<div class="menuBox">
<div class="menuBoxInner memberIcon">
            <div class="per_title">
            <h2><?php echo $page_title; ?></h2>
    </div>
    <div class="sub_section">
            <div class="btn-group">
    <a href="<?php echo MWB; ?>membership/index.php" class="btn btn-default"><?php echo __('Member List');
?></a>
    <a href="<?php echo MWB; ?>membership/index.php?action=detail" class="btn btn-default"><?php echo
__('Add New Member'); ?></a>
    <a href="<?php echo MWB; ?>membership/index.php?expire=true" class="btn btn-danger"><?php echo __('View
Expired Member'); ?></a>
            </div>
    <form name="search" action="<?php echo MWB; ?>membership/index.php" id="search" method="get"
class="form-inline"><?php echo __('Search'); ?>
                <input type="text" name="keywords" class="form-control col-md-3" /><?php if (isset($_GET['expire']))
{ echo '<input type="hidden" name="expire" value="true" />'; } ?>
                <input type="submit" id="doSearch" value="<?php echo __('Search'); ?>" class="s-btn btn btn-default"
/>
            </form>
            </div>
</div>
</div>
<?php
/* search form end */
/* main content */
if (isset($_POST['detail']) OR (isset($_GET['action']) AND $_GET['action'] == 'detail')) {
    if (!($can_read AND $can_write)) {
        die('<div class="errorBox">'.__('You don\'t have enough privileges to view this section').'</div>');
    }
    /* RECORD FORM */
    $itemID = $dbs->escape_string(trim(isset($_POST['itemID'])?$_POST['itemID']:''));
    $rec_q = $dbs->query("SELECT * FROM member WHERE member_id='$itemID'");
    $rec_d = $rec_q->fetch_assoc();
    // create new instance
    $form = new simbio_form_table_AJAX('mainForm',
$_SERVER['PHP_SELF'].'?'.$_SERVER['QUERY_STRING'], 'post');
    $form->submit_button_attr = 'name="saveData" value="'.__('Save').'" class="s-btn btn btn-default"';
    // form table attributes
    $form->table_attr = 'id="dataList" class="s-table table"';
    $form->table_header_attr = 'class="alterCell font-weight-bold"';
    $form->table_content_attr = 'class="alterCell2"';
    // edit mode flag set
    if ($rec_q->num_rows > 0) {
        $form->edit_mode = true;
        // record ID for delete process
        $form->record_id = $itemID;
        // form record title
        $form->record_title = $rec_d['member_name'];
        // submit button attribute
        $form->submit_button_attr = 'name="saveData" value="'.__('Update').'" class="s-btn btn btn-primary"';
        // custom field data query
        $_sql_rec_cust_q = sprintf('SELECT * FROM member_custom WHERE member_id=%d', $itemID);
        $rec_cust_q = $dbs->query($_sql_rec_cust_q);
        $rec_cust_d = $rec_cust_q->fetch_assoc();
    }
    /* Form Element(s) */
    if ($form->edit_mode) {
        // execute registered hook
        \SLiMS\Plugins::getInstance()->execute('membership_before_update', ['data' => api::member_load($dbs,
$itemID)]);
        // check if member expired
        $curr_date = date('Y-m-d');
        $compared_date = simbio_date::compareDates($rec_d['expire_date'], $curr_date);
        $is_expired = ($compared_date == $curr_date);
        $expired_message = '';
        if ($is_expired) {
            // extend membership
            $chbox_array[] = array('1', __('Extend'));
            $form->addCheckBox('extend', __('Extend Membership'), $chbox_array);
            $expired_message = '<strong class="text-danger">('.__('Membership Already Expired').')</strong>';
        }
    }
    // include custom fields file
    if (file_exists(MDLBS.'membership/member_custom_fields.inc.php')) {
        include MDLBS.'membership/member_custom_fields.inc.php';
    }
    // member code
    $str_input  = '<div class="container-fluid">';
    $str_input .= '<div class="row">';
    $str_input .= simbio_form_element::textField('text', 'memberID', $rec_d['member_id']??'', 'id="memberID"
onblur="ajaxCheckID(\''.SWB.'admin/AJAX_check_id.php\', \'member\', \'member_id\', \'msgBox\', \'memberID\')"
class="form-control col-4"');
    $str_input .= '<div id="msgBox" class="col mt-2"></div>';
    $str_input .= '</div>';
    $str_input .= '</div>';
    $form->addAnything(__('Member ID').'*', $str_input);
    // member name
    $form->addTextField('text', 'memberName', __('Member Name').'*', $rec_d['member_name']??'', 'class="form-
control" style="width: 50%;"');
    // member birth date
    $form->addDateField('birthDate', __('Birth Date').'*', $rec_d['birth_date']??'','class="form-control"');
    // member since date
    $form->addDateField('sinceDate', __('Member Since').'*', $rec_d['member_since_date']??date('Y-m-
d'),'class="form-control"');
    // member register date
    $form->addDateField('regDate', __('Register Date').'*', $rec_d['register_date']??date('Y-m-d'),'class="form-
control"');
    // member expire date
    if ($form->edit_mode) {
        $form->addDateField('expDate', __('Expiry Date').'*', $rec_d['expire_date']??'','class="form-control"');
    } else {
        $chbox_array[] = array('1', __('Auto Set'));
        $str_input = '<div>'.simbio_form_element::checkBox('extend', $chbox_array, '1').'</div>';
        $str_input .= '<div>'.simbio_form_element::dateField('expDate', $rec_d['expire_date']??'', 'class="form-
control"').'</div>';
        $form->addAnything(__('Expiry Date').'*', $str_input);
    }
    // member institution
    $form->addTextField('text', 'instName', __('Institution'), $rec_d['inst_name']??'', 'class="form-control"
style="width: 100%;"');
    // member type
        // get mtype data related to this record from database
        $mtype_query = $dbs->query("SELECT member_type_id, member_type_name FROM mst_member_type");
        $mtype_options = array();
        while ($mtype_data = $mtype_query->fetch_row()) {
            $mtype_options[] = array($mtype_data[0], $mtype_data[1]);
        }
    $form->addSelectList('memberTypeID', __('Membership Type').'*', $mtype_options,
$rec_d['member_type_id']??'','class="form-control col-4"');
    // member gender
    $gender_chbox[0] = array('1', __('Male'));
    $gender_chbox[1] = array('0', __('Female'));
    $form->addRadio('gender', __('Sex'), $gender_chbox, !empty($rec_d['gender'])?$rec_d['gender']:'0');
    // member address
    $form->addTextField('textarea', 'memberAddress', __('Address'), $rec_d['member_address']??'', 'rows="2"
class="form-control" style="width: 100%;"');
    // member postal
    $form->addTextField('text', 'memberPostal', __('Postal Code'), $rec_d['postal_code']??'', 'class="form-control"
style="width: 50%;"');
    // member mail address
    $form->addTextField('textarea', 'memberMailAddress', __('Mail Address'), $rec_d['member_mail_address']??'',
'rows="2" class="form-control" style="width: 100%;"');
    // member phone
    $form->addTextField('text', 'memberPhone', __('Phone Number'), $rec_d['member_phone']??'', 'class="form-
control" style="width: 50%;"');
    // member fax
    $form->addTextField('text', 'memberFax', __('Fax Number'), $rec_d['member_fax']??'', 'class="form-control"
style="width: 50%;"');
    // member pin
    $form->addTextField('text', 'memberPIN', __('Personal ID Number'), $rec_d['pin']??'', 'class="form-control"
style="width: 50%;"');
    // member notes
    $form->addTextField('textarea', 'memberNotes', __('Notes'), $rec_d['member_notes']??'', 'rows="2" class="form-
control" style="width: 100%;"');
    /**
     * Custom fields
     */
    if (isset($member_custom_fields)) {
        if (is_array($member_custom_fields) && $member_custom_fields) {
            foreach ($member_custom_fields as $fid => $cfield) {
            // custom field properties
            $cf_dbfield = $cfield['dbfield'];
            $cf_label = $cfield['label'];
            $cf_default = $cfield['default'];
            $cf_class = $cfield['class']??'';
            $cf_data = (isset($cfield['data']) && $cfield['data'] )?unserialize($cfield['data']):array();
            // get data field record
            if(isset($rec_cust_d[$cf_dbfield]) && @unserialize($rec_cust_d[$cf_dbfield]) !== false){
              $rec_cust_d[$cf_dbfield] = unserialize($rec_cust_d[$cf_dbfield]);
            }
            // custom field processing
            if (in_array($cfield['type'], array('text', 'longtext', 'numeric'))) {
              $cf_max = isset($cfield['max'])?$cfield['max']:'200';
              $cf_width = isset($cfield['width'])?$cfield['width']:'50';
              $form->addTextField( ($cfield['type'] == 'longtext')?'textarea':'text', $cf_dbfield, $cf_label,
$rec_cust_d[$cf_dbfield]??$cf_default, ' class="form-control '.$cf_class.'" style="width: '.$cf_width.'%;"
maxlength="'.$cf_max.'"');
            } else if ($cfield['type'] == 'dropdown') {
```

```
            $form->addSelectList($cf_dbfield, $cf_label, $cf_data, $rec_cust_d[$cf_dbfield]??$cf_default,' class="form-
control '.$cf_class.'"');
          } else if ($cfield['type'] == 'checklist') {
            $form->addCheckBox($cf_dbfield, $cf_label, $cf_data, $rec_cust_d[$cf_dbfield]??$cf_default,' class="form-
control '.$cf_class.'"');
          } else if ($cfield['type'] == 'choice') {
            $form->addRadio($cf_dbfield, $cf_label, $cf_data, $rec_cust_d[$cf_dbfield]??$cf_default,' class="form-
control '.$cf_class.'"');
          } else if ($cfield['type'] == 'date') {
            $form->addDateField($cf_dbfield, $cf_label, $rec_cust_d[$cf_dbfield]??$cf_default,' class="form-control
'.$cf_class.'"');
          }
          unset($cf_data);
          }
      }
    }
    // member is_pending
    $form->addCheckBox('isPending', __('Pending Membership'), array( array('1', __('Yes')) ),
$rec_d['is_pending']??'');
    // member photo
    $upper_dir  = '';
    $str_input  = '<div class="row">';
    $str_input .= '<div class="col-2">';
    $str_input .= '<div id="imageFilename" class="s-margin__bottom-1">';
    if (isset($rec_d['member_image']) && file_exists(IMGBS.'/persons/'.$rec_d['member_image'])) {
        $str_input .= '<a href="'.SWB.'images/persons/'.$rec_d['member_image'].'" class="openPopUp notAJAX"
title="'.__('Click to enlarge preview').'" width="300" height="400">';
        // $str_input .= '<img
src="'.$upper_dir.'../lib/minigalnano/createthumb.php?filename=images/persons/'.urlencode(($rec_d['member_image
']??'photo.png')).'&width=130" class="img-fluid" alt="Image cover">';
        $str_input .= '<img
src="'.$upper_dir.'../images/persons/'.urlencode(($rec_d['member_image']??'photo.png')).'?'.date('this').'"
class="img-fluid rounded" alt="Image cover">';
        $str_input .= '</a>';
        $str_input .= '<a href="'.MWB.'membership/index.php"
postdata="removeImage=true&mimg='.$itemID.'&img='.($rec_d['member_image']??'photo.png').'"
loadcontainer="imageFilename" class="s-margin__bottom-1 s-btn btn btn-danger btn-block rounded-0 makeHidden
removeImage">'.__('Remove Image').'</a>';
    }else{
        $str_input .= '<img src="'.SWB.'images/persons/person.png'.'?'.date('this').'" class="img-fluid rounded"
alt="Image cover">';
    }
    $str_input .= '</div>';
    $str_input .= '</div>';
    $str_input .= '<div class="custom-file col-4">';
    $str_input .= simbio_form_element::textField('file', 'image', '', 'class="custom-file-input"');
    $str_input .= '<label class="custom-file-label" for="customFile">Choose file</label>';
    $str_input .= '</div>';
    $str_input .= ' <div class="mt-2 ml-2">Maximum '.$sysconf['max_image_upload'].' KB</div>';
    $str_input .= '</div>';
    // $str_input = '<div id="imageFilename"><a href="'.SWB.'images/persons/'.$rec_d['member_image'].'"
class="openPopUp notAJAX"><strong>'.$rec_d['member_image'].'</strong></a> <a
href="'.MWB.'membership/index.php"
postdata="removeImage=true&mimg='.$itemID.'&img='.$rec_d['member_image'].'"
loadcontainer="imageFilename" class="makeHidden removeImage">'.__('REMOVE IMAGE').'</a></div>';
    // $str_input .= simbio_form_element::textField('file', 'image');
    // $str_input .= ' '.__('Maximum').' '.$sysconf['max_image_upload'].' KB';
    if ($sysconf['webcam'] !== false) {
      $str_input .= '<textarea id="base64picstring" name="base64picstring" style="display: none;"></textarea>';

      if ($sysconf['webcam'] == 'flex') {
```

```
      $str_input .= '<object id="flash_video" classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
height="280px" width="100%">';
      $str_input .= '<param name="src" value="'.SWB.'lib/flex/ShotSLiMSMemberPicture.swf"/>';
      $str_input .= '<embed name="src" src="'.SWB.'lib/flex/ShotSLiMSMemberPicture.swf" height="280px"
width="100%"/>';
      $str_input .= '</object>';
     }
     elseif ($sysconf['webcam'] == 'html5') {
      $str_input .= '<div class="makeHidden_">';
      $str_input .= '<p>'.__('or take a photo').'</p>';
      $str_input .= '<div class="form-inline">';
      $str_input .= '<div class="form-group pr-2">';
      $str_input .= '<button id="btn_load" class="btn btn-primary" onclick="loadcam(this)">'.__('Load
Camera').'</button>';
      $str_input .= '</div>';
      $str_input .= '<div class="form-group pr-2">';
      $str_input .= '<select class="form-control" onchange="aspect(this)"><option value="1">1x1</option><option
value="2" selected>2x3</option><option value="3">3x4</option></select>';
      $str_input .= '</div>';
      $str_input .= '<div class="form-group pr-2">';
      $str_input .= '<select class="form-control" id="cmb_format" onchange="if(pause){set();}"><option
value="png">PNG</option><option value="jpg">JPEG</option></select>';
      $str_input .= '</div>';
      $str_input .= '<div class="form-group pr-2">';
      $str_input .= '<button id="btn_pause" class="btn btn-primary" onclick="snapshot(this)"
disabled>'.__('Capture').'</button>';
      $str_input .= '</div>';
      $str_input .= '<div class="form-group pr-2">';
      $str_input .= '<button type="button" id="btn_reset" class="btn btn-danger"
onclick="resetvalue()">'.__('Reset').'</button>';
      $str_input .= '</div>';
      $str_input .= '</div>';
      $str_input .= '<div id="my_container" class="makeHidden_ mt-2" style="width: 400px; height: 300px; border:
1px solid #f4f4f4; position: relative;">';
      $str_input .= '<video id="my_vid" autoplay width="400" height="300" style="float: left; position: absolute; left:
10;"></video>';
      $str_input .= '<canvas id="my_canvas" width="400" height="300" style="float: left; position: absolute; left: 10;
visibility: hidden;"></canvas>';
      $str_input .= '<div id="my_frame" style="border: 1px solid #CCC; width: 160px; height: 240px; z-index: 2;
margin: auto; position: absolute; top: 0; bottom: 0; left: 0; right: 0;"></div></div>';
      $str_input .= '<canvas id="my_preview" width="160" height="240" style="width: 160px; height: 240px; border:
1px solid #f4f4f4; display: none;"></canvas>';
      }
   }
   $form->addAnything(__('Photo'), $str_input);
   // hidden username and password fields so that the password manager of the browser will not fill in the username in
the memberEmail and the password in the memberPasswd field
   $form->addTextField('text', 'dummyUserField', null, null, '');
   $form->addTextField('password', 'dummyPasswdField', null, null, '');
   echo '<style type="text/css">#simbioFormRowdummyPasswdField, #simbioFormRowdummyUserField {display:
none}</style>';
   // member email
   $form->addTextField('text', 'memberEmail', __('E-mail'), $rec_d['member_email']??'', 'class="form-control"
style="width: 40%;" class="form-control"');
   // member password
   $form->addTextField('password', 'memberPasswd', __('New Password'), null, 'class="form-control" style="width:
40%;" class="form-control" autocomplete="new-password"');
   // member password confirmation
   $form->addTextField('password', 'memberPasswd2', __('Confirm New Password'), null, 'class="form-control"
style="width: 40%;" class="form-control" autocomplete="new-password"');
   // edit mode messagge
   if ($form->edit_mode) {
        if (isset($rec_d['member_image'])) {
          if (file_exists(IMGBS.'persons/'.$rec_d['member_image'])) {
              echo '<div id="memberImage"><img
src="'.SWB.'images/persons/'.urlencode($rec_d['member_image']).'?'.date('his').'" alt="'.$rec_d['member_name'].'"
/></div>';
          }
        }
        echo '<div class="infoBox">
              <div>'.__('You are going to edit member data').' : <strong>'.$rec_d['member_name'].'</strong></div>
              <div>'.__('Last Updated').' '.date('d F Y h:i:s',strtotime($rec_d['last_update'])).' '.$expired_message.'</div>
              <div>'.__('Leave Password field blank if you don\'t want to change the password').'</div>';
        echo '</div>'."\n";
    }
    // print out the form object
    echo $form->printOut();
?>
<script type="text/javascript">
$(document).ready(function() {
    $('.removeImage').click(function (e) {
      if (confirm('Are you sure you want to permanently remove this image?')) {
        return true;
      } else {
        return false;
      }
    });
    $(document).on('change', '.custom-file-input', function () {
        let fileName = $(this).val().replace(/\\/g, '/').replace(/.*\//, '');
        $(this).parent('.custom-file').find('.custom-file-label').text(fileName);
    });
});
</script>
<?php
} else {
    /* MEMBERSHIP LIST */
    function showMemberImage($obj_db, $array_data){
      global $sysconf;
      $image = '../images/persons/photo.png';
      $_q = $obj_db->query('SELECT member_image,member_name,member_address,member_phone FROM member
WHERE member_id = "'.$array_data[0].'"');
      if(isset($_q->num_rows)){
        $_d = $_q->fetch_row();
        if($_d[0] != NULL){
          $image = file_exists(IMGBS.'/persons/'.$_d[0])?'../images/persons/'.$_d[0]:'../images/persons/photo.png';
        }
        $addr  = $_d[2]!=''?'<i class="fa fa-map-marker" aria-hidden="true"></i></i>&nbsp;'.$_d[2]:'';
        $phone = $_d[3]!=''?'<i class="fa fa-phone" aria-hidden="true"></i>&nbsp;'.$_d[3]:'';
      }
       $_output = '<div class="media">
                  <a href="'.$image.'" class="openPopUp notAJAX" title="'.$_d[1].'" width="300" height="400" >
                  <img class="mr-3 rounded" src="'.$image.'" alt="cover image" width="60"></a>
                  <div class="media-body">
                    <div class="title">'.$array_data[2].'</div>
                    <div class="sub">'.$phone.'</div>
                    <div class="sub">'.$addr.'</div>
                  </div>
                </div>';
       return $_output;
    }
    // table spec
    $table_spec = 'member AS m
        LEFT JOIN mst_member_type AS mt ON m.member_type_id=mt.member_type_id';
    // create datagrid
    $datagrid = new simbio_datagrid();
    if ($can_read AND $can_write) {
        $datagrid->setSQLColumn('m.member_id',
            'm.member_id AS \''.__('Member ID').'\'',
            'm.member_name AS \''.__('Member Name').'\'',
            'mt.member_type_name AS \''.__('Membership Type').'\'',
            'm.member_email AS \''.__('E-mail').'\'',
            'm.last_update AS \''.__('Last Updated').'\'');
            $datagrid->modifyColumnContent(2, 'callback{showMemberImage}');
    } else {
        $datagrid->setSQLColumn('m.member_id AS \''.__('Member ID').'\'',
            'm.member_name AS \''.__('Member Name').'\'',
            'mt.member_type_name AS \''.__('Membership Type').'\'',
            'm.member_email AS \''.__('E-mail').'\'',
            'm.last_update AS \''.__('Last Updated').'\'');
            $datagrid->modifyColumnContent(1, 'callback{showMemberImage}');
    }
    $datagrid->setSQLorder('member_name ASC');
    // is there any search
    $criteria = 'm.member_id IS NOT NULL ';
    if (isset($_GET['keywords']) AND $_GET['keywords']) {
       $keywords = $dbs->escape_string($_GET['keywords']);
       $criteria .= " AND (m.member_name LIKE '%$keywords%' OR m.member_id LIKE '%$keywords%') ";
    }
    if (isset($_GET['expire'])) {
        $criteria .= " AND TO_DAYS('".date('Y-m-d')."')>TO_DAYS(m.expire_date)";
    }
    $datagrid->setSQLCriteria($criteria);
    // set table and table header attributes
    $datagrid->icon_edit =
SWB.'admin/'.$sysconf['admin_template']['dir'].'/'.$sysconf['admin_template']['theme'].'/edit.gif';
    $datagrid->table_name = 'memberList';
    $datagrid->table_attr = 'id="dataList" class="s-table table"';
    $datagrid->table_header_attr = 'class="dataListHeader" style="font-weight: bold;"';
    // set delete proccess URL
    $datagrid->chbox_form_URL = $_SERVER['PHP_SELF'];
    // put the result into variables
    $datagrid_result = $datagrid->createDataGrid($dbs, $table_spec, 20, ($can_read AND $can_write));
    if ((isset($_GET['keywords']) AND $_GET['keywords']) OR isset($_GET['expire'])) {
        if (isset($_GET['expire'])) {
            echo '<div class="infoBox">';
            echo '<input type="button" value="'.__('Extend Selected Member(s)').'" onclick="javascript: if
(confirm(\''.__('Are you sure to EXTEND membership for selected members?').'\')) {
$(\'#mainContent\').simbioAJAX(\''.MWB.'membership/index.php?expire=1\', { method: \'post\', addData:
$(\'#memberList\').serialize() + \'&batchExtend=true\' } ); }" class="s-btn btn btn-primary" />';
            echo '</div>';
            if (isset($_GET['numExtended']) AND $_GET['numExtended'] > 0) {
                echo '<div class="infoBox mt-1">';
                echo '<strong>'.$_GET['numExtended'].'</strong> '.__('members extended!'); //mfc
                echo '</div>';
            }
        }
        if (isset($_GET['keywords']) AND $_GET['keywords']) {
            echo __('Found').' '.$datagrid->num_rows.' '.__('from your search with keyword').' :
"'.htmlentities($_GET['keywords']).'"'; //mfc
        }
        echo '</div>';
    }
    echo $datagrid_result;
}
```