

DAFTAR PUSTAKA

- Frieyadie, Sukmawati, A. H., & Nurajijah. (2020). Combination of the SAW and TOPSIS Method for Determining the Best Marketplace Recommendations. *Journal of Physics: Conference Series*, 1641(1). <https://doi.org/10.1088/1742-6596/1641/1/012004>
- Hanif, K. H., Yudhana, A., & Fadlil, A. (2020). Analisis Penilaian Guru Memakai Metode Visekriterijumsko Kompromisno Rangiranje (VIKOR). *Jurnal Ilmiah Mandala Education*, 6(1), 6–11. <https://doi.org/10.36312/jime.v6i1.1099>
- Harijanto, B., Tamara, N. A., & Ariyanto, Y. (2018). Pengembangan Aplikasi Pemilihan Kost Di Kota Malang Dengan Metode Ahp Dan Promethee. *Jurnal Informatika Polinema*, 4(3), 229. <https://doi.org/10.33795/jip.v4i3.212>
- Ismarmiaty, I., & Rizky, A. (2020). Sistem Pendukung Keputusan Perekrutan Karyawan PT. Cakra Mobilindo Menggunakan Metode Simple Additive Weighting. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 20(1), 117–128. <https://doi.org/10.30812/matrik.v20i1.827>
- Jadianan Parhusip. (2019). Penerapan Metode Analytical Hierarchy Process (AHP) Pada Desain Sistem Pendukung Keputusan Pemilihan Calon Penerima Bantuan Pangan Non Tunai (BPNT) Di Kota Palangka Raya. *Jurnal Teknologi Informasi Jurnal Keilmuan Dan Aplikasi Bidang Teknik Informatika*, 13(2), 18–29. <https://doi.org/10.47111/jti.v13i2.251>
- Kharis Hudaiby Hanif, Anton Yudhana, A. F. (2020). Analisis Penilaian Guru Memakai Metode Analityc Heararchy Process (AHP). *Seri Prosiding Seminar Nasional Dinamika Informatika*, 4(1), 186–189.
- Kustiyaningsih. (2011). *Sistem Informasi Perpustakaan Online Berbasis Web*. 5(2).
- Lambang, W. D. A. N., & Ii, B. A. B. (n.d.). *Tanfidz Keputusan Mukthamar XVI*.
- Lutfi, D. S. (2019). Sistem Pendukung Keputusan Penerimaan Dosen Baru Dengan Metode Simple Additive Weighting (Saw). *Jurnal Teknologi Informasi Dan*

Terapan, 4(1), 11–16. <https://doi.org/10.25047/jtit.v4i1.15>

Mahendra, G. S., & Sari, N. K. A. P. (2019). *Perancangan Sistem Pendukung Keputusan Dengan Metode Ahp-Vikor Dalam Penentuan Pengembangan Ekowisata Pedesaan (Decision Support System Design With Ahp-Vikor Method In Determination Of Rural Ecotourism Development)* (Issue November).

Mulyati, S. (2016). Penerapan Metode Simple Additive Weighting Untuk Penentuan Prioritas Pemasaran Kemasan Produk Bakso Sapi. *Jurnal Informatika*, 1(1), 33–37.

Nasional, S., & Teknik, P. (2019). *SENAPATI ke-10 Tahun 2019 “ Digital Education Entrepreneurship .” September.*

Ningsih, E., Dedih, & Supriyadi. (2017). Usaha Makanan Yang Tepat Menggunakan Weighted Product (WP) Berbasis Web. *ILKOM Jurnal Ilmiah*, 9(3), 244–254.

Resti, N. C. (2017). Penerapan Metode Simple Additive Weighting (SAW) pada Sistem Pendukung Keputusan Pemilihan Lokasi untuk Cabang Baru Toko Pakan UD. Indo Multi Fish. *Intensif*, 1(2), 102. <https://doi.org/10.29407/intensif.v1i2.839>

Sari, D. R., Windarto, A. P., Hartama, D., & Solikhun, S. (2018). Decision Support System for Thesis Graduation Recommendation Using AHP-TOPSIS Method. *Jurnal Teknologi Dan Sistem Komputer*, 6(1), 1–6. <https://doi.org/10.14710/jtsiskom.6.1.2018.1-6>

Sari, Y. K., Kartini, D., & Muliadi, M. (2018). Implementasi Algoritma Saw(Simple Additive Weighting) Dempster Shafer Pada Diagnosa Awal Postpartum Depression. *Computer Engineering, Science and System Journal*, 3(1), 1. <https://doi.org/10.24114/cess.v3i1.8332>

Sutinah, E. (2017). *Sistem Pendukung Keputusan Menggunakan Metode Profile Matching Dalam Pemilihan Salesman Terbaik*. 2(1), 29–42.

Umar, R., Fadlil, A., & Yuminah, Y. (2018). Sistem Pendukung Keputusan dengan

Metode AHP untuk Penilaian Kompetensi Soft Skill Karyawan. *Khazanah Informatika: Jurnal Ilmu Komputer Dan Informatika*, 4(1), 27. <https://doi.org/10.23917/khif.v4i1.5978>

Yuminah, Umar, R., & Fadlil, A. (2020). Analisis Metode Ahp Dan Promethee Pada Sistem Pendukung Keputusan Penilaian Kompetensi Soft Skills Karyawan Analysis of Ahp and Promethee Method on Decision Support System for Employee'S Soft Skills Competence Assesment. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 7(1), 27–36. <https://doi.org/10.25126/jtiik202071118>

Sugiyono. (2014) Metode Penelitian Pendekatan Kuantitatif., Bandung: Alfabeta

L

A

M

P

I

R

A

N

Content.php

```
<?php
if (!isset($_GET['page'])) {
    include 'page/dashboard.php';
} else {
    switch ($_GET['page']) {
        case 'dashboard':
            include 'page/dashboard.php';
            break;
        case 'alternatif':
            include 'page/data_alternatif.php';
            break;
        case 'kriteria':
            include 'page/data_kriteria.php';
            break;
        case 'nilai':
            include 'page/data_nilai.php';
            break;
        case 'seleksi':
            include 'page/data_seleksi.php';
            break;
        case 'pengguna':
            include 'page/data_pengguna.php';
            break;
        case 'backup':
            include 'page/data_backup.php';
            break;
        case 'ubah-password':
            include 'page/ubah_password.php';
            break;
    }
}
?>
```

Index.php

```
<?php
include "./config/library.php";
include "./config/koneksi.php";
session_start();

if(!$_SESSION['id_pengguna']){
    header("location:login");
}

$page=@$_GET['page'];
$id_pengguna=@$_SESSION['id_pengguna'];

opendb();
$sql=querydb("SELECT      *      FROM      saw_pengguna      WHERE
id_pengguna='$id_pengguna'");
$data=$sql->fetch_array();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>SPK Metode Simple Additive Weighting | Metode SAW</title>
    <link rel="icon" href="assets/images/ikon.png"/>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="author" content="Warung Koding">

    <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-
scalable=no" name="viewport">
    <link rel="stylesheet" href="assets/component/bootstrap/dist/css/bootstrap.min.css">
        <link    rel="stylesheet"    href="assets/component/font-awesome/css/font-
awesome.min.css">
            <link        rel="stylesheet"        href="assets/component/datatables.net-
bs/css/dataTables.bootstrap.min.css">
                <link rel="stylesheet" href="assets/component/select2/dist/css/select2.min.css">
                    <link rel="stylesheet" href="assets/dist/css/AdminLTE.min.css">
                        <link rel="stylesheet" href="assets/dist/css/skins/_all-skins.min.css">
                            <link                                rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,700,300i
tal,400italic,600italic">
</head>
```

```

<body class="hold-transition skin-yellow sidebar-mini">
  <div class="wrapper">
    <header class="main-header">
      <a href="." class="logo">
        <span class="logo-mini">SPK</span>
        <span class="logo-lg"><b>SPK</b> SAW</span>
      </a>
      <nav class="navbar navbar-static-top">
        <a href="#" class="sidebar-toggle" data-toggle="push-menu"
role="button"><span class="sr-only">Toggle navigation</span></a>
        <div class="navbar-custom-menu">
          <ul class="nav navbar-nav">
            <li class="dropdown user user-menu">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                
                <span class="hidden-xs"><?= $data['nama_pengguna'] ?></span>
              </a>
              <ul class="dropdown-menu">
                <li class="user-header">
                  
                  <p>
                    <?= $data['nama_pengguna'] ?>
                    <small>Sistem Pendukung Keputusan</small>
                    <small>Simple Additive Weighting</small>
                  </p>
                </li>
                <li class="user-footer">
                  <div class="form-group">
                    <a href="?page=ubah-password" class="btn btn-default pull-left"
title="Ubah Password">Password</a>
                    <a href="logout.php" class="btn btn-default pull-right"
onclick="return confirm('Apakah anda yakin ingin keluar?');">Logout</a>
                  </div>
                </li>
              </ul>
            </li>
          </ul>
        </div>
      </nav>

```

```

</header>
<aside class="main-sidebar">
  <section class="sidebar">
    <ul class="sidebar-menu" data-widget="tree">
      <li class="header">Menu</li>
      <li class="<?php if($page==" " || $page=="dashboard") { echo "active"; }
?>">
        <a href="."><i class="fa fa-dashboard"></i>
<span>Dashboard</span></a>
      </li>
      <li class="<?php if($page=="alternatif") { echo "active"; } ?>">
        <a href="?page=alternatif"><i class="fa fa-users"></i> <span>Data
Alternatif</span></a>
      </li>
      <li class="<?php if($page=="kriteria") { echo "active"; } ?>">
        <a href="?page=kriteria"><i class="fa fa-cubes"></i> <span>Data
Kriteria</span></a>
      </li>
      <li class="<?php if($page=="nilai" || $page=="seleksi" ) { echo "active"; }
?> treeview">
        <a href="#">
          <i class="fa fa-bar-chart"></i> <span>Seleksi Metode SAW</span>
          <span class="pull-right-container"><i class="fa fa-angle-left pull-
right"></i></span>
        </a>
        <ul class="treeview-menu">
          <li class="<?php if($page=="nilai") { echo "active"; } ?>"><a
href="?page=nilai"><i class="fa fa-check-square-o"></i> Nilai Alternatif</a></li>
          <li class="<?php if($page=="seleksi") { echo "active"; } ?>"><a
href="?page=seleksi"><i class="fa fa-check-square-o"></i> Hasil Perhitungan</a></li>
        </ul>
      </li>
      <li class="<?php if($page=="pengguna" || $page=="backup" ) { echo
"active"; } ?> treeview">
        <a href="#">
          <i class="fa fa-gear"></i> <span>Setting</span>
          <span class="pull-right-container"><i class="fa fa-angle-left pull-
right"></i></span>
        </a>
        <ul class="treeview-menu">
          <li class="<?php if($page=="pengguna") { echo "active"; } ?>"><a
href="?page=pengguna"><i class="fa fa-check-square-o"></i> Data Pengguna</a></li>
        </ul>

```

```

        </li>
        <li>
            <a href="logout.php" onclick="return confirm('Apakah anda yakin ingin
keluar?');"><i class="fa fa-power-off"></i> <span>Logout</span></a>
        </li>
    </ul>
</section>
</aside>
<div class="content-wrapper">
    <?php include "content.php"; ?>
</div>
<footer class="main-footer">
    <div class="pull-right hidden-xs">Aplikasi SPK Metode SAW</div>
    Copyright &copy; <?= date('Y'); ?> <a href="https://umb.ac.id/"
target="_blank">Umb</a>. All Right Reserved.
</footer>
</div>
<script src="assets/component/jquery/dist/jquery.min.js"></script>
<script src="assets/component/jquery-ui/jquery-ui.min.js"></script>
<script src="assets/component/bootstrap/dist/js/bootstrap.min.js"></script>
<script src="assets/component/datatables.net/js/jquery.dataTables.min.js"></script>
<script src="assets/component/datatables.net-bs/js/dataTables.bootstrap.min.js"></script>
<script src="assets/component/select2/dist/js/select2.full.min.js"></script>
<script>
    $(function () {
        $('select2').select2()
        $('#example1').DataTable()
    })
</script>
<script src="assets/dist/js/adminlte.min.js"></script>
<script src="assets/dist/js/pages/dashboard.js"></script>
</body>
</html>
<?php closedb(); ?>

```

Login.php

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Login Administrator | SPK Metode MOORA</title>
  <link rel="icon" href="assets/images/ikon.png">
  <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-
scalable=no" name="viewport">
  <link rel="stylesheet" href="assets/dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="assets/component/font-awesome/css/font-
awesome.min.css">
  <link rel="stylesheet" href="assets/dist/css/AdminLTE.min.css">
</head>
<body class="hold-transition login-page">
  <div class="login-box">
    <div class="login-box-body">
      <div align="center"></div><br>
      <div class="text-center">Login Administrator</div><br>
      <form action="login_periksa.php" method="post">
        <div class="form-group has-feedback">
          <input type="text" name="username" class="form-control"
placeholder="Username" required>
          <span class="fa fa-user form-control-feedback"></span>
        </div>
        <div class="form-group has-feedback">
          <input type="password" name="password" class="form-control"
placeholder="Password" required>
          <span class="fa fa-lock form-control-feedback"></span>
        </div>
        <div class="form-group has-feedback">
          <button type="submit" name="btn_login" class="btn btn-primary btn-
block"><span class="fa fa-sign-in"></span> Masuk</button>
        </div>
      </form>
    </div>
  </body>
</html>
```

Login_periksa.php

```
<?php
include("config/library.php");
include("config/koneksi.php");

opendb();
$username=$_POST['username'];
$password=md5($_POST['password']);

$hasil=querydb("SELECT * FROM saw_pengguna WHERE username='$username'
AND password='$password'");
$dt=$hasil->fetch_array();

if ($hasil->num_rows > 0) {
    session_start();
    $_SESSION['id_pengguna'] = $dt['id_pengguna'];
    $_SESSION['nama_pengguna'] = $dt['nama_pengguna'];
    $_SESSION['username'] = $dt['username'];
    $_SESSION['tipe'] = $dt['tipe'];
    header("location: ./");
} else {
    echo "<script language='JavaScript'>alert('Maaf, username atau password yang Anda
    masukkan salah. Silakan coba lagi.');
```

Koneksi.php

```
<?php
include "parser-php-version.php";

$dbhost = "localhost";
$dbname = "spk_saw";
$dbuser = "root";
$dbpassword = "";

$dbconnection = null;
function.opendb()
{
    global $dbhost, $dbuser, $dbpassword, $dbname, $dbconnection;
    $dbconnection = new mysqli($dbhost, $dbuser, $dbpassword, $dbname);
    if ($dbconnection->connect_error) {
        die("Koneksi gagal: " . $dbconnection->connect_error);
    }
}
function.closedb()
{
    global $dbconnection;
    $dbconnection->close();
}
function.querydb($query)
{
    global $dbconnection;
    $result = $dbconnection->query($query);
    if ($result === false) {
        $backtrace = debug_backtrace();
        $error_line = isset($backtrace[0]['line']) ? $backtrace[0]['line'] : 'Tidak dapat mengambil informasi baris.';
        die("<b>Warning</b>: Gagal melakukan query in <b>" . $dbconnection->error . "</b> on line <b>$error_line</b>");
    }
    return $result;
}
?>
```

Library.php

```
<?php
function tgl_indo($tgl){
    $tanggal = substr($tgl,8,2);
    $bulan = getBulan(substr($tgl,5,2));
    $tahun = substr($tgl,0,4);
    return $tanggal.' '.$bulan.' '.$tahun;
}
function getBulan($bln){
    switch ($bln){
        case 1:
            return "Januari";
            break;
        case 2:
            return "Februari";
            break;
        case 3:
            return "Maret";
            break;
        case 4:
            return "April";
            break;
        case 5:
            return "Mei";
            break;
        case 6:
            return "Juni";
            break;
        case 7:
            return "Juli";
            break;
        case 8:
            return "Agustus";
            break;
        case 9:
            return "September";
            break;
```

```
case 10:
    return "Oktober";
    break;
case 11:
    return "November";
    break;
case 12:
    return "Desember";
    break;
}
}
?>
```

Parser-php-version

<?php

```
# PHP MySQL to MySQLi migration shim library
#  purpose: Redefines deprecated or missing mysql_ functions and calls mysqli_
#           functions for PHP5.5+.
#  website: http://mysql.dotpointer.com/
#  author: Robert Klebe, dotpointer, dotpointer at gmail.com
#  licensing: Public domain, edit and share without permission.

# changelog
# 2013-09-24 xx:xx:xx - first version
# 2013-09-25 xx:xx:xx - updated with is_mysql_resource() and bugfixes
# 2013-09-26 19:23:00 - adding some more function wrappers
# 2013-10-24 23:16:41 - bugfix, mysql_connect used invalid error functions
# 2013-11-11 xx:xx:xx - updated mysql_result() contributed by marc17 (GitHub)
# 2014-02-16 xx:xx:xx - updated mysql_result() contributed by marc17 (GitHub)
# 2015-03-24 21:57:45 - updated is_mysql_resource() with suggestions from Colin
#                       McKinnon
# 2015-03-25 00:06:50 - updated is_mysql_resource() with suggestions from Colin
#                       McKinnon
# 2016-02-20 00:35:36 - checking that function error return values are the same as the
#                       original ones as suggested by Yaff Are
# 2016-02-23 21:30:57 - bugfix, mysql_connect halted execution upon invalid
#                       credentials
# 2016-02-23 21:58:43 - bugfix, mysql_create_db - quotation forbidden
# 2016-02-23 22:19:36 - bugfix, mysql_list_dbs - null when no link
# 2016-02-23 22:24:20 - bugfix, mysql_list_tables - quotation forbidden
# 2016-02-23 22:27:38 - bugfix, mysql_ping - null when no link
# 2016-02-23 22:47:44 - bugfix, mysql_info - null when no link
# 2016-02-23 22:59:36 - bugfix, mysql_errno, mysql_error - null when no link
# 2016-02-23 23:34:45 - adding mysql_db_name
# 2016-02-23 23:38:01 - bugfix, mysql_client_encoding - null when no link
# 2016-02-23 22:19:36 - bugfix, mysql_data_seek - null when invalid row number
# 2016-02-23 23:55:36 - bugfix, mysql_drop_db - quotation forbidden
# 2016-02-24 00:12:41 - bugfix, mysql_db_query - selected database could be
#                       bypassed
# 2016-02-24 00:38:04 - bugfix, mysql_fetch_field - null when invalid parameters
# 2016-02-24 01:02:44 - bugfix, mysql_fetch_lengths - null when invalid parameters
# 2016-02-24 02:17:27 - bugfix, mysql_field_flags returned invalid value, adding
#                       mysql_field_bitflags_to_flags
```

```

# 2016-02-24 02:35:15 - bugfix, mysql_field_seek - null when invalid parameters
# 2016-02-24 02:55:09 - bugfix, mysql_field_type - returned invalid value, adding
mysql_field_bittypes_to_types
# 2016-02-24 03:40:26 - bugfix, mysql_get_client_info, adding link argument
# 2016-02-24 03:43:55 - bugfix, mysql_get_host_info - null when invalid parameters
# 2016-02-24 04:10:19 - bugfix, mysql_get_proto_info - null when invalid parameters
# 2016-02-24 04:25:01 - bugfix, mysql_get_server_info - null when invalid parameters
# 2016-02-24 15:13:31 - bugfix, mysql_affected_rows - invalid value when invalid
parameters
# 2016-02-24 15:17:10 - bugfix, mysql_thread_i - null when invalid parameters
# 2016-02-24 17:47:23 - bugfix, mysql_insert_id - null when invalid parameters
# 2016-02-24 18:29:02 - adding mysql_list_processes
# 2016-02-24 19:21:21 - bugfix, mysql_num_fields - null when invalid parameters
# 2016-02-25 16:08:51 - cleanup
# 2016-02-25 18:08:54 - cleanup
# notes
# mysql constants are directly translated to mysqli, so the actual value may differ
# mysql_escape_string now takes the last used connection
# to check if an item is a (mysql) resource or a mysqli object
# note that all types of resources must get through as this is a
# generic replacement for is_resource()
function is_mysqli_or_resource($r) {
    # get the type of the variable
    switch(gettype($r)) {
        # if it is a resource - could be mysql, file handle etc...
        case 'resource':
            return true;
        # if it is an object - must be a mysqli object then
        case 'object':
            # is this an instance of mysqli?
            if ($r instanceof mysqli) {
                # make sure there is no connection error
                return !($r->connect_error);
            }
            # or is this an instance of a mysqli result?
            if ($r instanceof mysqli_result) {
                return true;
            }
        }
    }
}

```

```

        return false;
    # negative on all other variable types
    default:
        return false;
    }
# alias for is_mysqli_or_resource()
function is_mysql_resource($r) {
    return is_mysqli_or_resource($r);
}
# alias for is_mysqli_or_resource()
function is_generic_resource($r) {
    return is_mysqli_or_resource($r);
}
# to check if an item is a resource/object - replace is_resource with this
# old version, this will break if testing file handles too
function is_mysql_resource_old($result) {
    # first try to treat as resource if original mysql is loaded
    if (extension_loaded('mysql')) return is_resource($result);
    # or if mysqli is loaded, try to check object
    if (extension_loaded('mysqli')) return is_object($result);
    die('Fatal error, mysqli extension not loaded.');
```

```

}
# only do this is mysql extension is not there
if (!extension_loaded('mysql')) {
    # check if mysqli extension is loaded - its required as we rely on it
    if (!extension_loaded('mysqli')) die('Fatal error, mysqli extension not loaded.');
```

```

    # --- helper variables and constants
    # a list of connections, used to get the last one
    $mysql_links = array();
    # our own constants to reach default connection values in INI file
    define('MYSQL_DEFAULT_HOST', ini_get("mysql.default_host"));
    define('MYSQL_DEFAULT_USER', ini_get("mysql.default_user"));
    define('MYSQL_DEFAULT_PASSWORD', ini_get("mysql.default_password"));
    # --- MySQL constants (from PHP.net)
    # MySQL client constants
    define('MYSQL_CLIENT_COMPRESS', MYSQLI_CLIENT_COMPRESS);
    # Use compression protocol
    define('MYSQL_CLIENT_IGNORE_SPACE',
MYSQLI_CLIENT_IGNORE_SPACE); # Allow space after function names

```

```

define('MYSQL_CLIENT_INTERACTIVE',
MYSQLI_CLIENT_INTERACTIVE);    # Allow interactive_timeout seconds
                                # (instead of wait_timeout ) of
                                # inactivity before closing the connection.

define('MYSQL_CLIENT_SSL', MYSQLI_CLIENT_SSL);    # Use SSL
encryption. This flag is only
                                # available with version 4.x of the MySQL
                                # client library or newer. Version 3.23.x is
                                # bundled both with PHP 4 and Windows binaries
                                # of PHP 5.

# mysql_fetch_array() uses a constant for the different types of result arrays. The
following constants are defined:

# MySQL fetch constants

define('MYSQL_ASSOC', MYSQLI_ASSOC);    # Columns are returned into the
array having
                                # the fieldname as the array index.

define('MYSQL_BOTH', MYSQLI_BOTH); # Columns are returned into the array
having
                                # both a numerical index and the fieldname as
                                # the array index.

define('MYSQL_NUM', MYSQLI_NUM);    # Columns are returned into the array
having a
                                # numerical index to the fields. This index
                                # starts with 0, the first field in the result.

# --- helper functions -----
-----

# internal function to convert bitflags of mysqli to flags in text of mysql

# thanks to andre at koethur dot de at http://www.php.net/manual/en/mysqli-
result.fetch-fields.php#101828

function mysql_field_bitflags_to_flags($flags_num) {
    $flags = array();
    $constants = get_defined_constants(true);
    foreach ($constants['mysqli'] as $c => $n) {
        if (preg_match('/MYSQLI_(.*)_FLAG$/', $c, $m)) {
            if (!array_key_exists($n, $flags)) {
                $flags[$n] = $m[1];
            }
        }
    }
}

$result = array();
foreach ($flags as $n => $t) {
    if ($flags_num & $n) {

```

```

        $result[] = $t;
    }
}
return implode(' ', $result);
}

# function to convert bit-types of mysqli to types in text of mysql
# thanks to andre at koethur dot de at http://www.php.net/manual/en/mysqli-
result.fetch-fields.php#101828
function mysql_field_bittypes_to_types($type_id) {
    $types = array();
    $constants = get_defined_constants(true);
    foreach ($constants['mysqli'] as $c => $n) {
        if (preg_match('/^MYSQLI_TYPE_(.*)/', $c, $m)) {
            $types[$n] = $m[1];
        }
    }
}

return array_key_exists($type_id, $types)? $types[$type_id] : NULL;
}

# lib helper function - to ensure mysql link as mysqli always needs one
# but mysql takes last one
function mysql_ensure_link($link_identifier) {
    # no link specified
    if ($link_identifier === NULL) {
        global $mysql_links;
        # no connection at all - then go null
        if (!count($mysql_links)) return NULL;
        # get the last item of the array
        $last = end($mysql_links);
        # return the last stored link
        return $last['link'];
    }
    return $link_identifier;
}

# --- MySQL functions (from PHP.net) -----
-----

# mysql_affected_rows - Get number of affected rows in previous MySQL operation
# int mysql_affected_rows ([ resource $link_identifier = NULL ] )
# int mysqli_affected_rows ( mysqli $link )
function mysql_affected_rows($link_identifier = NULL) {

```

```

# mysql_affected_rows = -1 if the last query failed
# mysqli_affected_rows = -1 indicates that the query returned an error
$temp = mysqli_affected_rows(mysql_ensure_link($link_identifier));
if ($temp === NULL || $temp === false) {
    return -1;
}
return $temp;
}

# mysql_client_encoding - Returns the name of the character set
# string mysql_client_encoding ([ resource $link_identifier = NULL ])
# mysqli_character_set_name ( mysqli $link )
function mysql_client_encoding($link_identifier = NULL) {
    # note that mysqli_client_encoding ALSO is deprecated, so we cannot use that
    # mysql_client_encoding/mysqli_character_set_name = Returns the default
    # character set name for the current connection.
    $temp = mysqli_character_set_name(mysql_ensure_link($link_identifier));
    if ($temp === NULL) {
        return false;
    }
    return $temp;
}

# mysql_close - Close MySQL connection
# bool mysql_close ([ resource $link_identifier = NULL ])
# bool mysqli_close ( mysqli $link )
function mysql_close($link = NULL) {
    # mysql_close/mysqli_close = returns TRUE on success or FALSE on failure.
    global $mysql_links;
    $link = mysql_ensure_link($link);
    $thread_id = isset($link->thread_id) && is_numeric($link->thread_id) ? $link->thread_id : false;
    $result = mysqli_close($link);
    # did the removal succeed and we have thread id
    if ($result && $thread_id) {
        # walk the links
        foreach ($mysql_links as $k => $v) {
            # does this thread-id match the one we just removed?
            if ($v['thread_id'] === $thread_id) {

```

```

# then remove it from connection array
    array_splice($mysql_links, $k, 1);
    break;
}

}

# when connection already has been closed this error appears:
# Couldn't fetch mysqli in mysql.php on line xxx
# and this gives null instead of false
} else if ($result === null) {
    return false;
}

return $result;
}

# mysql_connect - Open a connection to a MySQL Server

# resource mysql_connect ([ string $server = ini_get("mysql.default_host") [, string
$username = ini_get("mysql.default_user") [, string $password =
ini_get("mysql.default_password") [, bool $new_link = false [, int $client_flags = 0 ]]]]
)

# mysqli mysqli_connect ([ string $host = ini_get("mysqli.default_host") [, string
$username = ini_get("mysqli.default_user") [, string $passwd =
ini_get("mysqli.default_pw") [, string $dbname = "" [, int $port =
ini_get("mysqli.default_port") [, string $socket = ini_get("mysqli.default_socket") ]]]]]
)

function mysql_connect($server = MYSQL_DEFAULT_HOST, $username =
MYSQL_DEFAULT_USER, $password = MYSQL_DEFAULT_PASSWORD,
$new_link = false, $client_flags = 0) {
    global $mysql_links;

    # no newlink but s/u/p matches prev ones-take last link
    if (!$new_link) {
        global $mysql_links;

        # are there prev links?
        if (count($mysql_links)) {
            # get the last one made
            $last = end($mysql_links);

            # does the s/u/p match last one?
            if ($server === $last['server'] && $username === $last['username'] &&
                $password === $last['password'] && is_resource($last['link'])) {
                # then take that
                return mysql_ensure_link(NULL);
            }
        }
    }
}

```

```

# try to connect using current credentials
$link = mysqli_connect($server,$username,$password,"");

if (mysqli_connect_errno()) {
    # printf("Connect failed: %s\n", mysqli_connect_error());
    # die();
    return false;
}

# store this
$mysql_links[] = array(
    'thread_id' => $link->thread_id,
    'server' => $server,
    'username' => $username,
    'password' => $password,
    'link' => $link
);
return $link;
}

# mysql_create_db - Create a MySQL database
# bool mysql_create_db ( string $database_name [, resource $link_identifier =
NULL] )

# CREATE DATABASE
function mysql_create_db($database_name, $link_identifier = NULL) {
    # mysql_create_db/mysql_query+CREATE DATABASE = false on error
    return mysql_query('CREATE DATABASE
'.mysql_real_escape_string($database_name), $link_identifier);
}

# mysql_data_seek - Move internal result pointer
# bool mysql_data_seek ( resource $result , int $row_number )
# bool mysqli_data_seek ( mysqli_result $result , int $offset )
function mysql_data_seek($result, $row_number) {
    # mysql_data_seek/mysqli_data_seek = false on error
    $temp = mysqli_data_seek($result, $row_number);
    if ($temp === NULL) {
        return false;
    }
    return true;
}

```

```

# mysql_db_name - Retrieves database name from the call to # mysql_list_dbs
# string mysql_db_name ( resource $result , int $row [, mixed $field = NULL ] )
# SELECT DATABASE()
function mysql_db_name($result , $row, $field = NULL) {
    # return mysql_query('SELECT DATABASE()',
mysql_ensure_link($link_identifier));

    # null does not fit mysql_result
    $field = $field === null ? 0 : $field;

    return mysql_result($result, $row, $field);
}

# mysql_db_query - Selects a database and executes a query on it
# resource mysql_db_query ( string $database , string $query [, resource
$link_identifier = NULL ] )
# mysqli_select_db() then the query
function mysql_db_query($database, $query, $link_identifier = NULL) {
    # mysql_db_query = false on error, mysql_query+sql = false on error
    if (mysqli_select_db($database, $link_identifier) !== true) {
        return false;
    }
    return mysql_query($query, $link_identifier);
}

# mysql_drop_db - Drop (delete) a MySQL database
# bool mysql_drop_db ( string $database_name [, resource $link_identifier = NULL ]
)
# DROP DATABASE
function mysql_drop_db($database_name, $link_identifier = NULL) {
    # mysql_drop_db = false on error, mysql_query+DROP DATABASE = false on
error
    return mysql_query('DROP DATABASE
'.mysql_real_escape_string($database_name), $link_identifier);
}

# mysql_errno -Returns the numerical value of the error message from previous
MySQL operation
# int mysql_errno ([ resource $link_identifier = NULL ] )
# int mysqli_errno ( mysqli $link )
function mysql_errno($link_identifier = NULL) {
    # mysql_errno/mysqli_errno = returns a number, 0 if no error
    $temp = mysqli_errno (mysql_ensure_link($link_identifier));

```

```

if ($temp === NULL) {
    return false;
}
return $temp;
}

# mysql_error - Returns the text of the error message from previous MySQL
operation
# string mysql_error ([ resource $link_identifier = NULL ] )
# string mysqli_error ( mysqli $link )
function mysql_error($link_identifier = NULL) {
    # mysql_error/mysqli_error = returns empty string on no error
    $temp = mysqli_error(mysql_ensure_link($link_identifier));
    if ($temp === NULL) {
        return false;
    }
    return $temp;
}

# mysql_escape_string - Escapes a string for use in a # mysql_query
# string mysql_escape_string ( string $unescaped_string )
# string mysqli::real_escape_string ( string $escapestr )
function mysql_escape_string($unescaped_string) {
    # mysql_escape_string = returns the escaped string
    # mysql_real_escape_string = returns FALSE on error
    return mysql_real_escape_string($unescaped_string);
}

# mysql_fetch_array - Fetch a result row as an associative array, a numeric array, or
both
# array mysql_fetch_array ( resource $result [, int $result_type = MYSQL_BOTH ] )
# mixed mysqli_fetch_array ( mysqli_result $result [, int $resulttype =
MYSQLI_BOTH ] )
function mysql_fetch_array($result, $result_type = MYSQL_BOTH) {

    # mysql_fetch_array = Returns an array of strings that corresponds to the fetched
row, or FALSE if there are no more rows

    # mysqli_fetch_array = Returns an array of strings that corresponds to the fetched
row or NULL if there are no more rows in resultset

    # store the result in a temporarily array
    $temp = mysqli_fetch_array($result, $result_type);

    # is the result null?

```

```

if ($temp === NULL) {
    # then return false as the old function did
    return false;
}
return $temp;
}

# mysql_fetch_assoc - Fetch a result row as an associative array
# array mysql_fetch_assoc ( resource $result )
# array mysqli_fetch_assoc ( mysqli_result $result )
function mysql_fetch_assoc ($result) {
    # mysql_fetch_assoc = returns FALSE if there are no more rows
    # mysqli_fetch_assoc = returns NULL if there are no more rows in resultset
    $temp = mysqli_fetch_assoc($result);
    # is the result null?
    if ($temp === NULL) {
        # then return false as the old function did
        return false;
    }
    return $temp;
}

# mysql_fetch_field - Get column information from a result and return as an object
# object mysql_fetch_field ( resource $result [, int $field_offset = 0 ] )
# object mysqli_fetch_field ( mysqli_result $result ) - but field_offset is missing
function mysql_fetch_field($result, $field_offset = NULL) {
    # if field offset is specified
    if (is_numeric($field_offset)) {
        # then seek to that
        mysqli_field_seek($result, $field_offset);
    }
    $temp = mysqli_fetch_field($result);

    if ($temp === NULL) {
        return false;
    }
    return $temp;
}

```

```

# mysql_fetch_lengths - Get the length of each output in a result
# array mysql_fetch_lengths ( resource $result )
# array mysqli_fetch_lengths ( mysqli_result $result )
function mysql_fetch_lengths($result) {
    # mysql_fetch_lengths/mysqli_fetch_lengths = FALSE on error
    $temp = mysqli_fetch_lengths($result);
    if ($temp === NULL) {
        return false;
    }
    return $temp;
}

# mysql_fetch_object - Fetch a result row as an object
# object mysql_fetch_object ( resource $result [, string $class_name [, array $params
]])
# object mysqli_fetch_object ( mysqli_result $result [, string $class_name [, array
$params ]) )
function mysql_fetch_object ($result, $class_name=NULL, $params=NULL) {

    # mysql_fetch_object = FALSE if there are no more rows
    # mysqli_fetch_object = NULL if there are no more rows in resultset

    if ($class_name !== NULL && $params !== NULL) {
        $temp = mysqli_fetch_object($result, $class_name, $params);
    } else if ($class_name !== NULL) {
        $temp = mysqli_fetch_object($result, $class_name);
    } else {
        $temp = mysqli_fetch_object($result);
    }
    # is the result null?
    if ($temp === NULL) {
        # then return false as the old function did
        return false;
    }
    return $temp;
}

# mysql_fetch_row - Get a result row as an enumerated array
# array mysql_fetch_row ( resource $result )
# mixed mysqli_fetch_row ( mysqli_result $result )
function mysql_fetch_row ($result) {

```

```

# mysql_fetch_row = FALSE if there are no more rows
# mysqli_fetch_row = NULL if there are no more rows in result set

$temp = mysqli_fetch_row($result);

# is the result null?
if ($temp === NULL) {
    # then return false as the old function did
    return false;
}
return $temp;
}

# mysql_field_flags - Get the flags associated with the specified field in a result
# string mysql_field_flags ( resource $result , int $field_offset )
# mysqli_fetch_field_direct() [flags]
# -> object mysqli_fetch_field_direct ( mysqli_result $result , int $fieldnr )
function mysql_field_flags($result, $field_offset) {
    # mysql_field_flags = FALSE on failure
    # mysqli_fetch_field_direct = FALSE if no field information for specified fieldnr
    is available
    $tmp = mysqli_fetch_field_direct($result, $field_offset);
    if (!is_object($tmp)) return false;
    $tmp = (array)$tmp;
    return isset($tmp['flags']) ? mysql_field_bitflags_to_flags($tmp['flags']) : false;
}

# mysql_field_len - Returns the length of the specified field
# int mysql_field_len ( resource $result , int $field_offset )
# mysqli_fetch_field_direct() [length]
# -> object mysqli_fetch_field_direct ( mysqli_result $result , int $fieldnr )
function mysql_field_len($result, $field_offset) {
    # mysql_field_len = FALSE on failure
    # mysqli_fetch_field_direct = FALSE if no field information for specified fieldnr
    is available
    $tmp = mysqli_fetch_field_direct($result, $field_offset);
    if (!is_object($tmp)) return false;
    $tmp = (array)$tmp;
    return isset($tmp['length']) ? $tmp['length'] : false;
}

```

```

# mysql_field_name - Get the name of the specified field in a result
# string mysql_field_name ( resource $result , int $field_offset )
# mysqli_fetch_field_direct() [name] or [orgname]
# -> object mysqli_fetch_field_direct ( mysqli_result $result , int $fieldnr )
function mysql_field_name($result, $field_offset) {
    # mysql_field_name = FALSE on failure
    # mysqli_fetch_field_direct = FALSE if no field information for specified fieldnr
    # is available
    $tmp = mysqli_fetch_field_direct($result, $field_offset);
    if (!is_object($tmp)) return false;
    $tmp = (array)$tmp;
    return isset($tmp['name']) ? $tmp['name'] : false;
}

# mysql_field_seek - Set result pointer to a specified field offset
# bool mysql_field_seek ( resource $result , int $field_offset )
# bool mysqli_field_seek ( mysqli_result $result , int $fieldnr )
function mysql_field_seek($result, $field_offset) {
    # mysql_field_seek/mysqli_field_seek = FALSE on failure
    $temp = mysqli_field_seek($result, $field_offset);
    if ($temp === NULL) {
        return false;
    }
    return $temp;
}

# mysql_field_table - Get name of the table the specified field is in
# string mysql_field_table ( resource $result , int $field_offset )
# mysqli_fetch_field_direct() [table] or [orgtable]
# -> object mysqli_fetch_field_direct ( mysqli_result $result , int $fieldnr )
function mysql_field_table($result, $field_offset) {
    # mysql_field_table = error return value not defined
    # mysqli_fetch_field_direct = FALSE if no field information for specified fieldnr
    # is available
    $tmp = mysqli_fetch_field_direct($result, $field_offset);
    if (!is_object($tmp)) return false;
    $tmp = (array)$tmp;
    return isset($tmp['table']) ? $tmp['table'] : false;
}

# mysql_field_type - Get the type of the specified field in a result

```

```

# string mysql_field_type ( resource $result , int $field_offset )
# mysqli_fetch_field_direct() [type]
# -> object mysqli_fetch_field_direct ( mysqli_result $result , int $fieldnr )
function mysql_field_type($result, $field_offset) {
    # mysql_field_type = error return value not defined
    # mysqli_fetch_field_direct = FALSE if no field information for specified fieldnr
    is available
    $tmp = mysqli_fetch_field_direct($result, $field_offset);
    if (!is_object($tmp)) return false;
    $tmp = (array)$tmp;
    return isset($tmp['type']) ? mysql_field_bittypes_to_types($tmp['type']) : false;
}

# mysql_free_result - Free result memory
# bool mysql_free_result ( resource $result )
# void mysqli_free_result ( mysqli_result $result )
function mysql_free_result($result) {
    # mysql_free_result = FALSE on failure
    # mysqli_free_result = No value is returned.
    mysqli_free_result($result);
    # note that mysqli does not return any boolean, so we do it
    return true;
}

# mysql_get_client_info - Get MySQL client info
# string mysql_get_client_info ( void )
# string mysqli_get_client_info ( mysqli $link )
function mysql_get_client_info($link_identifier = null) {
    # mysql_get_client_info/mysqli_get_client_info = not defined what is returned on
    error
    # note that mysql does not have a link argument while mysqli does
    return mysqli_get_client_info(mysql_ensure_link($link_identifier));
}

# mysql_get_host_info - Get MySQL host info
# string mysql_get_host_info ([ resource $link_identifier = NULL ] )
# string mysqli_get_host_info ( mysqli $link )
function mysql_get_host_info ($link_identifier = NULL) {
    # mysql_get_host_info = FALSE on failure
    # mysqli_get_host_info = error return value not defined
    $temp = mysqli_get_host_info(mysql_ensure_link($link_identifier));

```

```

if ($temp === NULL) {
    return false;
}
return $temp;
}

# mysql_get_proto_info - Get MySQL protocol info
# int mysql_get_proto_info ([ resource $link_identifier = NULL ] )
# int mysqli_get_proto_info ( mysqli $link )
function mysql_get_proto_info($link_identifier = NULL) {
    # mysql_get_proto_info = FALSE on failure
    # mysqli_get_proto_info = error return value not defined
    $temp = mysqli_get_proto_info(mysql_ensure_link($link_identifier));
    if ($temp === NULL) {
        return false;
    }
    return $temp;
}

# mysql_get_server_info - Get MySQL server info
# string mysql_get_server_info ([ resource $link_identifier = NULL ] )
# string mysqli_get_server_info ( mysqli $link )
function mysql_get_server_info($link_identifier = NULL) {
    # mysql_get_server_info = FALSE on failure
    # mysqli_get_server_info = error return value not defined
    $temp = mysqli_get_server_info(mysql_ensure_link($link_identifier));
    if ($temp === NULL) {
        return false;
    }
    return $temp;
}

# mysql_info - Get information about the most recent query
# string mysql_info ([ resource $link_identifier = NULL ] )
# string mysqli_info ( mysqli $link )
function mysql_info($link_identifier = NULL) {
    # mysql_info = FALSE on failure
    # mysqli_info = returns empty string on failure
    $temp = mysqli_info(mysql_ensure_link($link_identifier));
    if ($temp === NULL) {

```

```

        return false;
    }
    return $temp;
}

# mysql_insert_id - Get the ID generated in the last query
# int mysql_insert_id ([ resource $link_identifier = NULL ] )
# mixed mysqli_insert_id ( mysqli $link )
function mysql_insert_id($link_identifier = NULL) {
    # mysql_insert_id = FALSE if no MySQL connection was established
    # mysqli_insert_id = error value not defined
    $temp = mysqli_insert_id(mysql_ensure_link($link_identifier));
    if ($temp === null) {
        return false;
    }
    return $temp;
}

# mysql_list_dbs - List databases available on a MySQL server
# resource mysql_list_dbs ([ resource $link_identifier = NULL ] )
# SQL Query: SHOW DATABASES
function mysql_list_dbs ($link_identifier = NULL) {
    global $mysql_list_dbs_cache;
    # mysql_list_dbs/mysql_query = FALSE on failure
    $temp = mysql_query('SHOW DATABASES',
mysql_ensure_link($link_identifier));

    $mysql_list_dbs_cache = $temp;

    # when no working link is passed we get null
    if ($temp === NULL) {
        return false;
    }
    return $temp;
}

# mysql_list_fields - List MySQL table fields
# resource mysql_list_fields ( string $database_name , string $table_name [, resource
$link_identifier = NULL ] )
# SQL Query: SHOW COLUMNS FROM sometable
function mysql_list_fields ($database_name, $table_name, $link_identifier = NULL)
{

```

```

        # mysql_list_fields/mysql_query = FALSE on failure
        return mysql_query('SHOW COLUMNS FROM
'.mysql_real_escape_string($database_name).'.'.mysql_real_escape_string($table_name
).'', mysql_ensure_link($link_identifier));
    }

# mysql_list_processes - List MySQL processes
# resource mysql_list_processes ([ resource $link_identifier = NULL ] )
# mysqli_thread_id()
function mysql_list_processes($link_identifier = NULL) {
    # mysql_list_processes = FALSE on failure
    $temp = mysql_query("SHOW PROCESSLIST",
mysql_ensure_link($link_identifier));
    if ($temp === null) {
        return false;
    }
    return $temp;
}

# mysql_list_tables - List tables in a MySQL database
# resource mysql_list_tables ( string $database [, resource $link_identifier = NULL ]
)
# SQL Query: SHOW TABLES FROM sometable
function mysql_list_tables ($database_name, $table_name, $link_identifier = NULL)
{
    # mysql_list_tables/mysql_query = FALSE on failure
    return mysql_query('SHOW TABLES FROM
'.mysql_real_escape_string($database_name), mysql_ensure_link($link_identifier));
}

# mysql_num_fields - Get number of fields in result
# int mysql_num_fields ( resource $result )
# int mysqli_field_count ( mysqli $link )
function mysql_num_fields ($result) {
    # mysql_num_fields/mysqli_fetch_fields = FALSE on failure
    # mysql takes a result, where mysqli takes link and takes the most recent query
    # so instead we fetch all the fields and then count that
    $tmp = mysqli_fetch_fields($result);
    if ($tmp === null) {
        return false;
    }
    return count($tmp);
}

```

```

# mysql_num_rows - Get number of rows in result
# int mysql_num_rows ( resource $result )
# int mysqli_num_rows ( mysqli_result $result )
function mysql_num_rows($result) {
    # mysql_num_rows = FALSE on failure
    # mysqli_num_rows = error return value not defined
    return mysqli_num_rows($result);
}

# mysql_pconnect - Open a persistent connection to a MySQL server
# resource mysql_pconnect ([ string $server = ini_get("mysql.default_host") [, string
$username = ini_get("mysql.default_user") [, string $password =
ini_get("mysql.default_password") [, int $client_flags = 0 ]]] )
# mysqli_connect() with p: host prefix
function mysql_pconnect($server = MYSQL_DEFAULT_HOST, $username =
MYSQL_DEFAULT_USER, $password = MYSQL_DEFAULT_PASSWORD,
$client_flags = 0) {
    # mysql_pconnect/mysqli_connect = FALSE on error
    return mysql_connect('p:'. $server, $username, $password, true, $client_flags);
}

# mysql_ping - Ping a server connection or reconnect if there is no connection
# bool mysql_ping ([ resource $link_identifier = NULL ] )
# bool mysqli_ping ( mysqli $link )
function mysql_ping($link_identifier = NULL) {
    # mysql_ping/mysqli_ping = FALSE on error
    $temp = mysqli_ping(mysql_ensure_link($link_identifier));
    if ($temp === NULL) {
        return false;
    }
    return $temp;
}

# mysql_query - Send a MySQL query
# resource mysql_query ( string $query [, resource $link_identifier = NULL ] )
# mixed mysqli_query ( mysqli $link , string $query [, int $resultmode =
MYSQLI_STORE_RESULT ] )
function mysql_query ($query, $link_identifier = NULL) {
    # mysql_query/mysqli_query = FALSE on error
    return mysqli_query(mysql_ensure_link($link_identifier), $query);
}

```

mysql_real_escape_string - Escapes special characters in a string for use in an SQL statement

string mysql_real_escape_string (string \$unescape_string [, resource \$link_identifier = NULL])

string mysqli_real_escape_string (mysqli \$link , string \$escapestr)

function mysql_real_escape_string(\$unescape_string, \$link_identifier = NULL) {

mysql_real_escape_string = FALSE on error

mysqli_real_escape_string = error return value not defined

return mysqli_real_escape_string(mysqli_ensure_link(\$link_identifier), \$unescape_string);

}

mysql_result - Get result data

string mysql_result (resource \$result , int \$row [, mixed \$field = 0])

no equivalent function exists in mysqli - mysqli_data_seek() in conjunction with mysqli_field_seek() and mysqli_fetch_field()

function mysql_result(\$result , \$row , \$field = 0) {

mysql_result = FALSE on failure

try to seek position, returns false on failure

if (mysqli_data_seek(\$result, \$row) === false) return false;

\$row = mysqli_fetch_array(\$result);

if (\$row === NULL || !isset(\$row[\$field])) return false;

return \$row[\$field];

}

mysql_select_db - Select a MySQL database

bool mysql_select_db (string \$database_name [, resource \$link_identifier = NULL])

function mysql_select_db (\$database_name, \$link_identifier = NULL) {

mysql_select_db/mysqli_select_db = FALSE on failure

return mysqli_select_db(mysqli_ensure_link(\$link_identifier), \$database_name);

}

mysql_set_charset - Sets the client character set

bool mysql_set_charset (string \$charset [, resource \$link_identifier = NULL])

bool mysqli_set_charset (mysqli \$link , string \$charset)

function mysql_set_charset(\$charset, \$link_identifier = NULL) {

mysql_set_charset/mysqli_set_charset = FALSE on failure

return mysqli_set_charset(mysqli_ensure_link(\$link_identifier), \$charset);

}

```

# mysql_stat - Get current system status
# string mysql_stat ([ resource $link_identifier = NULL ] )
# string mysqli_stat ( mysqli $link )
function mysql_stat($link_identifier = NULL) {
    # mysql_stat = NULL on error
    # mysqli_stat = FALSE on error
    $temp = mysqli_stat(mysql_ensure_link($link_identifier));
    if ($temp === FALSE) {
        return NULL;
    }
    return $temp;
}

# mysql_tablename - Get table name of field
# string mysql_tablename ( resource $result , int $i )
# no mysqli equivalent exists - SHOW TABLES [FROM db_name] [LIKE 'pattern']
function mysql_tablename ($result, $i) {
    # return mysql_query('SHOW COLUMNS FROM
    "'.mysql_real_escape_string($database_name).'.mysql_real_escape_string($table_name
    )."', mysql_ensure_link($link_identifier));
    return mysql_result($result, $i);
}

# mysql_thread_id - Return the current thread ID
# int mysql_thread_id ([ resource $link_identifier = NULL ] )
# int mysqli_thread_id ( mysqli $link )
function mysql_thread_id($link_identifier = NULL) {
    # mysql_thread_id = FALSE on failure
    # mysqli_thread_id = no error return value defined
    $temp = mysqli_thread_id(mysql_ensure_link($link_identifier));
    if ($temp === NULL) {
        return false;
    }
    return $temp;
}
}
?>

```

Logout.php

```
<?php
session_start();
session_destroy();
header("Location: ./");
exit;
?>
```