

//Attendance_model.php

<?php

class Attendance_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get attendance by logID

*/

function get_attendance(\$logID)

{

return \$this->db->get_where('attendance',array('logID'=>\$logID))->row_array();

}

//get specific attendance log to time in.

function getAttendance(\$tutorID,\$dateNow)

{

\$where = "tutorID = '". \$tutorID. "' AND timeIn LIKE '". \$dateNow. "%";

return \$this->db->get_where('attendance',\$where)->row_array();

}

//list all tutors attendance

function get_list(\$term,\$sy,\$params = array())

{

```

$this->db->from('attendance a');
$this->db->join('tutors t', 't.tutorID = a.tutorID');
$this->db->join('users u', 'u.userID = t.userID');
$this->db->where('tutorType', 'Honor Scholar');
$this->db->where('term', $term);
$this->db->where('schoolYr', $sy);
$this->db->order_by('u.lastName', 'asc');
if(isset($params) && !empty($params))
{
    $this->db->limit($params['limit'], $params['offset']);
}
return $this->db->get()->result_array();
}

```

//specific tutor attendance list

```

function get_tutor_attendance_list($tutorID,$term,$sy ){
    $this->db->from('attendance');
    $this->db->where('tutorID', $tutorID);
    $this->db->where('term',$term);
    $this->db->where('schoolYr', $sy);
    return $this->db->get()->result_array();
}

```

/*

* Get all attendance count

*/

```

function get_all_attendance_count()

```

```

{

```

```

    $this->db->from('attendance');

```

```
    return $this->db->count_all_results();  
}
```

```
// count all attendance of a tutor
```

```
function get_tutor_attendance_count($tutorID,$term,$sy)  
{  
    $this->db->from('attendance');  
    $this->db->where('tutorID', $tutorID);  
    $this->db->where('term',$term);  
    $this->db->where('schoolYr', $sy);  
    return $this->db->count_all_results();  
}
```

```
/*
```

```
 * Get all attendance
```

```
*/
```

```
function get_all_attendance($params = array())  
{  
    $this->db->order_by('logID', 'desc');  
    if(isset($params) && !empty($params))  
    {  
        $this->db->limit($params['limit'], $params['offset']);  
    }  
    return $this->db->get('attendance')->result_array();  
}
```

```
/*
```

```
* function to add new attendance
*/
function add_attendance($params)
{
    $this->db->insert('attendance',$params);
    return $this->db->insert_id();
}

/*
* function to update attendance
*/
function update_attendance($logID,$params)
{
    $this->db->where('logID',$logID);
    return $this->db->update('attendance',$params);
}
}
```

//Auditlog_model.php

<?php

class Auditlog_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

//get auditlogs and users

function get_joinedaudit()

{

\$this->db->from('auditlogs a');

\$this->db->join('users u', 'u.userID = a.userID');

\$this->db->order_by('timeStamp','desc');

return \$this->db->get()->result_array();

}

/*

* function to add new auditlog

*/

function add_auditlog(\$params)

{

\$this->db->insert('auditlogs',\$params);

return \$this->db->insert_id();

}

}

//Faculty_model

<?php

class Faculty_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get faculty by facultyID

*/

function get_faculty(\$facultyID)

{

return \$this->db->get_where('faculty',array('facultyID'=>\$facultyID))->row_array();

}

/*

* Get all faculty count

*/

function get_all_faculty_count()

{

\$this->db->from('faculty');

return \$this->db->count_all_results();

}

/*

* Get all faculty

```

*/
function get_all_faculty($params = array())
{
    $this->db->order_by('facultyID', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    return $this->db->get('faculty')->result_array();
}

```

```

/*
* function to add new faculty
*/
function add_faculty($params)
{
    $this->db->insert('faculty',$params);
    return $this->db->insert_id();
}

```

```

/*
* function to update faculty
*/
function update_faculty($facultyID,$params)
{
    $this->db->where('facultyID',$facultyID);
    return $this->db->update('faculty',$params);
}

```

```
/*  
 * function to delete faculty  
 */  
function archive_faculty($facultyID)  
{  
    $this->db->where('facultyID',$facultyID);  
    return $this->db->update('faculty',$params);  
}  
}
```


//Feedback_model.php

<?php

class Feedback_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get feedback by feedbackID

*/

function get_feedback(\$feedbackID)

{

return \$this->db->get_where('feedbacks',array('feedbackID'=>\$feedbackID))->row_array();

}

/*

* Get all feedbacks count

*/

function get_all_feedbacks_count()

{

\$this->db->from('feedbacks');

return \$this->db->count_all_results();

}

/*

* Get all feedbacks

```

*/
function get_all_feedbacks($params = array())
{
    $this->db->order_by('feedbackID', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    return $this->db->get('feedbacks')->result_array();
}

/*
 * Get pending feedbacks count
 */
function pending_feedbacks_count()
{
    $where = 'ts.dateTimeEnd IS NOT NULL';
    $this->db->join('tutorialsessions ts', 'ts.tutorialNo = f.tutorialNo');
    $this->db->where($where);
    $this->db->from('feedbacks f');
    return $this->db->count_all_results();
}

/*
 * Get pending tutee feedbacks
 */
function pending_feedbacks($params = array())
{
    $where = 'ts.dateTimeEnd IS NOT NULL';

    $this->db->select('f.*, ts.tutorialNo, ts.tutorID, u.lastName, u.firstName, s.subjectCode,
ts.dateTimeEnd');

```

```

$this->db->join('tutorialsessions ts', 'ts.tutorialNo = f.tutorialNo', 'right');

$this->db->join('subjects s', 'ts.subjectID = s.subjectID');

$this->db->join('tutors tu', 'tu.tutorID = ts.tutorID', 'left');

$this->db->join('users u', 'tu.userID = u.userID', 'left');

$this->db->where($where);

$this->db->order_by('feedbackID', 'desc');

if(isset($params) && !empty($params))
{
    $this->db->limit($params['limit'], $params['offset']);
}

return $this->db->get('feedbacks f')->result_array();
}

/*
 * function to add new feedback
 */
function add_feedback($params)
{
    $this->db->insert('feedbacks',$params);

    return $this->db->insert_id();
}

/*
 * function to update feedback
 */
function update_feedback($feedbackID,$params)
{
    $this->db->where('feedbackID',$feedbackID);

    return $this->db->update('feedbacks',$params);
}
}

```

//Login_model.php

<?php

class Login_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get user by username and password

*/

function get_user(\$params = array())

{

return \$this->db->get_where('users',array('username'=>\$params['username']))->row_array();

}

function getuser(\$username)

{

return \$this->db->get_where('users',array('username'=>\$username))->row_array();

}

}

//Program_model.php

<?php

class Program_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get program by programID

*/

function get_program(\$programID)

{

return \$this->db->get_where('programs',array('programID'=>\$programID))->row_array();

}

/*

* Get all programs count

*/

function get_all_programs_count()

{

\$this->db->from('programs');

return \$this->db->count_all_results();

}

/*

* Get all programs

```

*/
function get_all_programs($params = array())
{
    $this->db->order_by('programID', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    return $this->db->get('programs')->result_array();
}

```

```

/*
* function to add new program
*/
function add_program($params)
{
    $this->db->insert('programs',$params);
    return $this->db->insert_id();
}

```

```

/*
* function to update program
*/
function update_program($programID,$params)
{
    $this->db->where('programID',$programID);
    return $this->db->update('programs',$params);
}

```

```
/*  
 * function to delete program  
 */  
function delete_program($programID)  
{  
    return $this->db->delete('programs',array('programID'=>$programID));  
}  
}
```

//Programcourse_model.php

<?php

class Programcourse_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get programcourse by refNo

*/

function get_programcourse(\$refNo)

{

return \$this->db->get_where('programcourses',array('refNo'=>\$refNo))->row_array();

}

/*

* Get all programcourses count

*/

function get_all_programcourses_count()

{

\$this->db->from('programcourses');

return \$this->db->count_all_results();

}

/*

* Get all programcourses


```

*/
function get_all_programcourses($params = array())
{
    $this->db->order_by('refNo', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    return $this->db->get('programcourses')->result_array();
}

```

```

/*
* function to add new programcourse
*/
function add_programcourse($params)
{
    $this->db->insert('programcourses',$params);
    return $this->db->insert_id();
}

```

```

/*
* function to update programcourse
*/
function update_programcourse($refNo,$params)
{
    $this->db->where('refNo',$refNo);
    return $this->db->update('programcourses',$params);
}

```

```
/*  
 * function to delete programcourse  
 */  
function delete_programcourse($refNo)  
{  
    return $this->db->delete('programcourses',array('refNo'=>$refNo));  
}  
}
```

//Timeblock_model.php

<?php

class Timeblock_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get timeblock by timeblockID

*/

function get_timeblock(\$timeblockID)

{

return \$this->db->get_where('timeblocks',array('timeblockID'=>\$timeblockID))->row_array();

}

/*

* Get all timeblocks count

*/

function get_all_timeblocks_count()

{

\$this->db->from('timeblocks');

return \$this->db->count_all_results();

}

/*

* Get all timeblocks

*/

```

function get_all_timeblocks($params = array())
{
    $this->db->order_by('timeblockID', 'asc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    return $this->db->get('timeblocks')->result_array();
}

/*
 * Get all timeblocks
 */
function get_timeblock_where($params = array())
{
    $this->db->order_by('timeblockID', 'asc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    return $this->db->get('timeblocks')->result_array();
}

function get_available_timeblock($param)
{
    $data = $this->db->get_where('timeblocks', array('status'=>$param));
    return $data->result_array();
}

/*

```

```
* function to add new timeblock
*/
function add_timeblock($params)
{
    $this->db->insert('timeblocks',$params);
    return $this->db->insert_id();
}

/*
* function to update timeblock
*/
function update_timeblock($timeblockID,$params)
{
    $this->db->where('timeblockID',$timeblockID);
    return $this->db->update('timeblocks',$params);
}

}
```

//Term_model.php

<?php

class Term_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

function get_current_term()

{

\$currentMonth = date('m');

if(\$currentMonth>8 && \$currentMonth<=12)

{

\$term=1;

\$sy=(date('Y'))."-".(date('Y')+1);

}

elseif (\$currentMonth>=1 && \$currentMonth<5) {

\$term=2;

\$sy=(date('Y')-1)."-".(date('Y'));

}

elseif (\$currentMonth>=5 && \$currentMonth<8) {

\$term=3;

\$sy=(date('Y')-1)."-".(date('Y'));

}

return array('term'=>\$term,'sy'=>\$sy);

}

function term_dissected(\$term)

```
{

    $month = date('m');
    $curr_yr = date('y');

    if($term = 1)
    {
        $start = date('Y-m-d H:i:s', strtotime($curr_yr.'-9-01 00:00:00'));
        $end = date('Y-m-d H:i:s', strtotime($curr_yr.'-12-31 00:00:00'));
    }
    elseif ($term = 2) {
        $start = date('Y-m-d H:i:s', strtotime($curr_yr . '-1-01 00:00:00'));
        $end = date('Y-m-d H:i:s', strtotime($curr_yr . '-4-28 00:00:00'));
    }
    elseif ($term = 3) {
        $start = date('Y-m-d H:i:s', strtotime($curr_yr . '-5-01 00:00:00'));
        $end = date('Y-m-d H:i:s', strtotime($curr_yr . '-8-31 00:00:00'));
    }

    return array('start'=>$start, 'end'=>$end);

}
}
```

//Tutee_model.php

<?php

class Tutee_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get tutee by tuteeID

*/

function get_tutee(\$tuteeID)

{

return \$this->db->get_where('tutees',array('tuteeID'=>\$tuteeID))->row_array();

}

/*

* Get all tutees count

*/

function get_all_tutees_count()

{

\$this->db->from('tutees');

return \$this->db->count_all_results();

}

/*

* Get all tutees for term


```

*/
function count_active_tutees($start, $end)
{
    $this->db->from('tutees t');
    $this->db->join('tutorialsessions ts', 'tuteeID');
    $this->db->where('ts.dateTimeEnd >=', $start);
    $this->db->where('ts.dateTimeEnd <=', $end);
    return $this->db->count_all_results();
}

/*
 * Get all tutees
 */
function get_all_tutees($params = array())
{
    $this->db->order_by('tuteeID', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    return $this->db->get('tutees')->result_array();
}

/*
 * function to add new tutee
 */
function add_tutee($params)
{
    $this->db->insert('tutees', $params);
}

```

```
        return $this->db->insert_id();
    }

    /**
     * function to update tutee
     */
    function update_tutee($tuteeID,$params)
    {
        $this->db->where('tuteeID',$tuteeID);
        return $this->db->update('tutees',$params);
    }

    /**
     * function to delete tutee
     */
    function delete_tutee($tuteeID)
    {
        return $this->db->delete('tutees',array('tuteeID'=>$tuteeID));
    }
}
```

//Tutor_model.php

<?php

class Tutor_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get tutor by tutorID

*/

function get_tutor(\$tutorID)

{

return \$this->db->get_where('tutors',array('tutorID'=>\$tutorID))->row_array();

}

/*

* Get tutor by userID

*/

function get_tutor_userID(\$userID)

{

return \$this->db->get_where('tutors',array('userID'=>\$userID))->row_array();

}

//Get tutor by userid

function get_tutorID(\$userID){

return \$this->db->get_where('tutors',array('userID'=>\$userID))->row_array();

}

```

/*
 * Get all tutors count
 */
function get_all_tutors_count()
{
    $this->db->from('tutors');
    return $this->db->count_all_results();
}

/*
 * Get all active tutors count
 */
function count_active_tutors($term)
{
    $this->db->from('tutors t');
    $this->db->join('tutorschedules ts', 'tutorID');
    $this->db->where('ts.term', $term['term']);
    $this->db->where('ts.schoolYr', $term['sy']);
    return $this->db->count_all_results();
}

/*
 * Get all tutors
 */
function get_all_tutors($params = array())
{
    $this->db->from('tutors t');
    $this->db->join('users u', 'u.userID = t.userID');
    $this->db->join('tutorstatus s', 's.statusID = t.statusID');
    $this->db->join('students st', 'st.userID = t.userID', 'left');

```

```

$this->db->join('programs p', 'p.programID = st.programID','left');
$this->db->order_by('t.tutorID', 'asc');
if(isset($params) && !empty($params))
{
    // $this->db->limit($params['limit'], $params['offset']);
    parent::__construct();
}

/*
 * Get tutor by tutorID
 */
function get_tutor($tutorID)
{
    return $this->db->get_where('tutors',array('tutorID'=>$tutorID))->row_array();
}

/*
 * Get tutor by userID
 */
function get_tutor_userID($userID)
{
    return $this->db->get_where('tutors',array('userID'=>$userID))->row_array();
}

//Get tutor by userid
function get_tutorID($userID){
    return $this->db->get_where('tutors',array('userID'=>$userID))->row_array();
}

```

```

}

/*
 * Get all tutors count
 */
function get_all_tutors_count()
{
    $this->db->from('tutors');
    return $this->db->count_all_results();
}

/*
 * Get all tutors
 */
function get_all_tutors($params = array())
{
    $this->db->from('tutors t');
    $this->db->join('users u', 'u.userID = t.userID');
    $this->db->order_by('t.tutorID', 'asc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    return $this->db->get()->result_array();
}

//get all honor scholars tutor
function get_honor_scholars()
{
    $this->db->from('tutors t');

```

```
$this->db->join('users u', 'u.userID = t.userID');  
$this->db->where('tutorType', 'Honor Scholar');  
$this->db->order_by('t.tutorID', 'asc');  
return $this->db->get()->result_array();  
}
```

```
/*  
 * function to add new tutor  
 */  
function add_tutor($params)  
{  
    $this->db->insert('tutors',$params);  
    return $this->db->insert_id();  
}
```

```
/*  
 * function to update tutor  
 */  
function update_tutor($tutorID,$params)  
{  
    $this->db->where('tutorID',$tutorID);  
    return $this->db->update('tutors',$params);  
}
```

```
/*  
 * function to delete tutor  
 */  
function delete_tutor($tutorID)  
{
```

```

        return $this->db->delete('tutors',array('tutorID'=>$tutorID));
    }

    return $this->db->get()->result_array();
}

/*
 * Get all tutors by term
 */
function tutors_by_term($params = array())
{
    $this->db->select('t.*, stu.studentNo, pr.programCode, u.lastName, u.firstName, s.status,
ts.dayofweek, tb.timeStart, tb.timeEnd');

    $this->db->from('tutors t');

    $this->db->join('users u', 'u.userID = t.userID');

    $this->db->join('students stu', 'u.userID = stu.userID');

    $this->db->join('tutorstatus s', 't.statusID = s.statusID');

    $this->db->join('tutorschedules ts', 'ts.tutorID = t.tutorID');

    $this->db->join('timeblocks tb', 'tb.timeblockID = ts.timeblockID');

    $this->db->join('programs pr', 'stu.programID = pr.programID', 'left');

    $this->db->order_by('t.tutorID', 'asc');

    return $this->db->get()->result_array();
}

/*
 * function to add new tutor
 */
function add_tutor($params)
{
    $this->db->insert('tutors',$params);
}

```



```
    return $this->db->insert_id();  
}
```

```
/*  
 * function to update tutor  
 */  
function update_tutor($tutorID,$params)  
{  
    $this->db->where('tutorID',$tutorID);  
    return $this->db->update('tutors',$params);  
}
```

```
/*  
 * function to delete tutor  
 */  
function delete_tutor($tutorID)  
{  
    return $this->db->delete('tutors',array('tutorID'=>$tutorID));  
}  
}
```

//Tutorexpertise_model.php

<?php

class Tutorexpertise_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get tutorexpertise by expertiselD

*/

function get_tutorexpertise(\$expertiselD)

{

return \$this->db->get_where('tutorexpertise',array('expertiselD'=>\$expertiselD))->row_array();

}

/*

* Get tutorexpertise by tutorID

*/

function tutorexpertise_by_tutorID(\$tutorID)

{

\$this->db->select('s.subjectCode');

\$this->db->join('subjects s', 's.subjectID = te.subjectID');

\$this->db->order_by('subjectCode', 'asc');

return \$this->db->get_where('tutorexpertise te',array('tutorID'=>\$tutorID))->result_array();

}

/*

```

* Get all tutorexperitise count
*/
function get_all_tutorexpertise_count()
{
    $this->db->from('tutorexpertise');
    return $this->db->count_all_results();
}

/*
* Get all tutorexperitise
*/
function get_all_tutorexpertise($params = array())
{
    $this->db->order_by('expertiseID', 'desc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    return $this->db->get('tutorexpertise')->result_array();
}

/*
* function to add new tutorexperitise
*/
function add_tutorexpertise($params)
{
    $this->db->insert('tutorexpertise',$params);
    return $this->db->insert_id();
}

```

```
/*
 * function to update tutorexperitise
 */
function update_tutorexpertise($expertiseID,$params)
{
    $this->db->where('expertiseID',$expertiseID);
    return $this->db->update('tutorexpertise',$params);
}

/*
 * function to delete tutorexperitise
 */
function delete_tutorexpertise($expertiseID)
{
    return $this->db->delete('tutorexpertise',array('expertiseID'=>$expertiseID));
}
}
```

//Tutorialchecklist_model.php

<?php

class Tutorialchecklist_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get tutorialchecklist by checklistID

*/

function get_tutorialchecklist(\$tutorialNo)

{

return \$this->db->get_where('tutorialchecklists',array('tutorialNo'=>\$tutorialNo))->result_array();

}

/*

* Get all tutorialchecklists count

*/

function count_tutNo_list(\$tutorialNo)

{

\$this->db->from('tutorialchecklists');

\$this->db->where('tutorialNo', \$tutorialNo);

return \$this->db->count_all_results();

}

/*

```
* Get all tutorialchecklists count
```

```
*/
```

```
function get_all_tutorialchecklists_count()
```

```
{
```

```
    $this->db->from('tutorialchecklists');
```

```
    return $this->db->count_all_results();
```

```
}
```

```
/*
```

```
* Get all tutorialchecklists
```

```
*/
```

```
function get_all_tutorialchecklists($params = array())
```

```
{
```

```
    $this->db->order_by('checklistID', 'desc');
```

```
    if(isset($params) && !empty($params))
```

```
    {
```

```
        $this->db->limit($params['limit'], $params['offset']);
```

```
    }
```

```
    return $this->db->get('tutorialchecklists')->result_array();
```

```
}
```

```
/*
```

```
* function to add new tutorialchecklist
```

```
*/
```

```
function add_tutorialchecklist($params)
```

```
{
```

```
    $this->db->insert('tutorialchecklists',$params);
```

```
    return $this->db->insert_id();
```

```
}
```

```

/*
 * function to update tutorialchecklist
 */
function update_tutorialchecklist($checklistID,$params)
{
    $this->db->where('checklistID',$checklistID);
    return $this->db->update('tutorialchecklists',$params);
}

/*
 * function to delete tutorialchecklist
 */
function delete_tutorialchecklist($tutorialNo)
{
    return $this->db->delete('tutorialchecklists',array('tutorialNo'=>$tutorialNo));
}
}

```

//tutorialsession_model.php

<?php

class Tutorialsession_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

function get_user_tutorialsessions_count()

{

\$this->db->join('tutees tee', 't.tuteeID = tee.tuteeID');

\$this->db->join('users utee', 'tee.userID = utee.userID');

\$this->db->where('utee.userID', \$_SESSION['userID']);

\$this->db->from('tutorialsessions t');

return \$this->db->count_all_results();

}

function count_userpending_sessions()

{

\$this->db->from('tutorialsessions t');

\$this->db->join('tutees tees', 't.tuteeID = tees.tuteeID');

\$this->db->join('users ut', 'ut.userID = tees.userID');

\$this->db->where('ut.userID', \$_SESSION['userID']);

\$this->db->where('t.status', "Pending");

return \$this->db->count_all_results();

}


```
function count_userapproved_sessions()
{
    $this->db->from('tutorialsessions t');
    $this->db->join('tutees tees', 't.tuteeID = tees.tuteeID');
    $this->db->join('users ut', 'ut.userID = tees.userID');
    $this->db->where('ut.userID', $_SESSION['userID']);
    $this->db->where('t.status', "Approved");
    return $this->db->count_all_results();
}
```

```
function count_tutorall_sessions()
{
    $this->db->from('tutorialsessions t');
    $this->db->join('tutors ta', 'ta.tutorID = t.tutorID', 'left');
    $this->db->join('users ua', 'ua.userID = ta.tutorID');
    $this->db->where('ua.userID', $_SESSION['userID']);
    return $this->db->count_all_results();
}
```

```
function count_tutorunstarted_sessions()
{
    $this->db->from('tutorialsessions t');
    $this->db->join('tutors ta', 'ta.tutorID = t.tutorID', 'left');
    $this->db->join('users ua', 'ua.userID = ta.tutorID');
    $this->db->where('ua.userID', $_SESSION['userID']);
    $this->db->where('t.dateTimeStart !=', null);
    return $this->db->count_all_results();
}
```

```
function count_tutorfinished_sessions()
{
    $this->db->from('tutorialsessions t');
    $this->db->join('tutors ta', 'ta.tutorID = t.tutorID', 'left');
    $this->db->join('users ua', 'ua.userID = ta.tutorID');
    $this->db->where('ua.userID', $_SESSION['userID']);
    $this->db->where('t.dateTimeEnd', null);
    return $this->db->count_all_results();
}
```

```
function count_adminall()
{
    $this->db->from('tutorialsessions t');
    return $this->db->count_all_results();
}
```

```
function count_adminpending()
{
    $this->db->from('tutorialsessions t');
    $this->db->where('t.status', "Pending");
    return $this->db->count_all_results();
}
```

```
function count_adminfeedback()
{
    $this->db->from('tutorialsessions t');
    $this->db->join('feedbacks f', 't.tutorialNo = f.tutorialNo');
    $this->db->where('f.feedback !=', null);
    return $this->db->count_all_results();
}
```

```
}
```

```
/*
```

```
 * Get tutorialsession by tutorialNo
```

```
*/
```

```
function get_tutorialsession($tutorialNo)
```

```
{
```

```
    return $this->db->get_where('tutorialsessions',array('tutorialNo'=>$tutorialNo))->row_array();
```

```
}
```

```
/*
```

```
 * Get all tutorialsessions count
```

```
*/
```

```
function get_all_tutorialsessions_count()
```

```
{
```

```
    $this->db->from('tutorialsessions');
```

```
    return $this->db->count_all_results();
```

```
}
```

```
/*
```

```
 * Get all tutorialsessions count
```

```
*/
```

```
function count_term_sessions($start, $end)
```

```
{
```

```
    $this->db->from('tutorialsessions ts');
```

```
    $this->db->where('ts.dateTimeEnd >=', $start);
```

```
    $this->db->where('ts.dateTimeEnd <=', $end);
```

```
    return $this->db->count_all_results();
```

```
}
```

```

/*
 * Get all requested sessions count
 */
function count_req_sessions($start, $end)
{
    $this->db->from('tutorialsessions ts');
    $this->db->where('ts.dateTimeRequested >=', $start);
    $this->db->where('ts.dateTimeRequested <=', $end);
    return $this->db->count_all_results();
}

```

```

/*
 * count sessions today
 */
function count_sessions_today($start, $end)
{
    $this->db->from('tutorialsessions ts');
    $this->db->where('ts.dateTimeRequested >=', $start);
    $this->db->where('ts.dateTimeRequested <=', $end);
    return $this->db->count_all_results();
}

```

```

/*
 * Get all tutorialsessions
 */
function get_all_tutorialsessions($params = array())
{

```

```
$this->db->select('t.*, utee.lastName uteeLN, utee.firstName uteeFN, utee.username uteeUN,
ur.lastName urLN, ur.firstName urFN, ua.lastName uaLN, ua.firstName uaFN, s.subjectCode,
tsr.dayofweek tsrdow, tbr.timeStart tbrTS, tbr.timeEnd tbrTE');
```

```
$this->db->from('tutorialsessions t');
```

```
$this->db->join('subjects s', 't.subjectID = s.subjectID');
```

```
$this->db->join('tutees tee', 't.tuteeID = tee.tuteeID');
```

```
$this->db->join('users utee', 'tee.userID = utee.userID');
```

```
$this->db->join('tutors tr', 'tr.tutorID = t.previousTutorID');
```

```
$this->db->join('tutors ta', 'ta.tutorID = t.tutorID', 'left');
```

```
$this->db->join('users ur', 'ur.userID = tr.tutorID');
```

```
$this->db->join('users ua', 'ua.userID = ta.tutorID', 'left');
```

```
$this->db->join('tutorschedules tsr', 'tsr.tutorScheduleID = t.tutorScheduleID');
```

```
$this->db->join('timeblocks tbr', 'tbr.timeblockID = tsr.timeblockID');
```

```
$this->db->order_by('tutorialNo', 'asc');
```

```
if(isset($params) && !empty($params))
```

```
{
```

```
    $this->db->limit($params['limit'], $params['offset']);
```

```
}
```

```
return $this->db->get()->result_array();
```

```
}
```

```
/*
```

```
* Get all tutorialsessions
```

```
*/
```

```
function get_user_tutorialsessions($params = array())
```

```
{
```

```
    $this->db->select('t.*, utee.lastName uteeLN, utee.firstName uteeFN, utee.username uteeUN,
ur.lastName urLN, ur.firstName urFN, ua.lastName uaLN, ua.firstName uaFN, s.subjectCode,
tsr.dayofweek tsrdow, tbr.timeStart tbrTS, tbr.timeEnd tbrTE');
```

```

$this->db->from('tutorialsessions t');

$this->db->join('subjects s', 't.subjectID = s.subjectID', 'left');

// tutees

$this->db->join('tutees tee', 't.tuteeID = tee.tuteeID', 'left');

$this->db->join('users utee', 'tee.userID = utee.userID', 'left');

// /tutees

// previous tutor

$this->db->join('tutors tr', 'tr.tutorID = t.previousTutorID', 'left');

$this->db->join('users ur', 'ur.userID = tr.userID', 'left');

// /previous tutor

// assigned tutor

$this->db->join('tutors ta', 'ta.tutorID = t.tutorID', 'left');

$this->db->join('users ua', 'ua.userID = ta.userID', 'left');

// /assigned tutor

$this->db->join('tutorschedules tsr', 'tsr.tutorScheduleID = t.tutorScheduleID', 'left');

$this->db->join('timeblocks tbr', 'tbr.timeblockID = tsr.timeblockID', 'left');

$this->db->where('utee.userID', $_SESSION['userID']);


$this->db->order_by('tutorialNo', 'asc');

if(isset($params) && !empty($params))
{
    $this->db->limit($params['limit'], $params['offset']);
}

return $this->db->get()->result_array();
}

/*
* function to add new tutorialsession
*/

```

```
function add_tutorialsession($params)
```

```
{
```

```
    $this->db->insert('tutorialsessions',$params);
```

```
    return $this->db->insert_id();
```

```
}
```

```
/*
```

```
 * function to update tutorialsession
```

```
*/
```

```
function update_tutorialsession($tutorialNo,$params)
```

```
{
```

```
    $this->db->where('tutorialNo',$tutorialNo);
```

```
    return $this->db->update('tutorialsessions',$params);
```

```
}
```

```
/*
```

```
 * function to delete tutorialsession
```

```
*/
```

```
function delete_tutorialsession($tutorialNo)
```

```
{
```

```
    return $this->db->delete('tutorialsessions',array('tutorialNo'=>$tutorialNo));
```

```
}
```

```
/*
```

```
 * Get all subjects by date and tutorschedule
```

```
*/
```

```
function subjects_by_date($dayofweek)
```

```
{
```

```
    $this->db->from('tutorschedules ts');
```

```

$this->db->join('tutorexpertise te', 'ts.tutorID = te.tutorID');
$this->db->join('subjects s', 's.subjectID = te.subjectID');
$this->db->where('ts.dayofweek', $dayofweek);
return $this->db->get()->result_array();
}

```

```

function get_tutors($subjectID)
{
    $where = "tutsess.dateTimeRequested != (SELECT DATE_FORMAT(dateTimeRequested,'%d-%m-%Y') as dateTimeRequested FROM tutorialsessions WHERE status ='Approved')";
    $this->db->from('tutorschedules ts');
    $this->db->join('tutorexpertise te', 'ts.tutorID = te.tutorID');
    $this->db->join('tutors tu', 'ts.tutorID = tu.tutorID');
    $this->db->join('users u', 'tu.userID = u.userID');
    $this->db->join('tutorialsessions tutsess', 'tutsess.tutorID = tu.tutorID');
    $this->db->join('subjects s', 's.subjectID = te.subjectID');
    $this->db->where('s.subjectID', $subjectID);
    $this->db->where($where);
    return $this->db->get()->result_array();
}

```

```

/*

```

```

* Get all days and timeblocks by subject

```

```

*/

```

```

function tutorschedules_by_subject($subjectID)
{
    $this->db->from('tutorschedules ts');
    $this->db->join('timeblocks tb', 'tb.timeblockID = ts.timeblockID');
}

```



```

$this->db->join('tutorexpertise te', 'ts.tutorID = te.tutorID');
$this->db->join('tutors tu', 'ts.tutorID = tu.tutorID');
$this->db->join('users u', 'tu.userID = u.userID');
$this->db->join('subjects s', 's.subjectID = te.subjectID');
$this->db->where('s.subjectID', $subjectID);
return $this->db->get()->result_array();
}

/*
 * Get tutors by tutorsched
 */
function tutors_by_tutorscheds($subjectID)
{
    $this->db->from('tutorschedules ts');
    $this->db->join('timeblocks tb', 'tb.timeblockID = ts.timeblockID');
    $this->db->join('tutorexpertise te', 'ts.tutorID = te.tutorID');

    $this->db->join('subjects s', 's.subjectID = te.subjectID');
    $this->db->where('s.subjectID', $subjectID);
    return $this->db->get()->result_array();
}

/*
 * Get occupied days filtered by subject and tutorsched availability
 */
function tsessions_by_subj($subjectID, $tutorScheduleID)
{
    $this->db->select('tses.tutorialNo, tses.dateTimeRequested');
    $this->db->from('tutorialsessions tses');

```

```

$this->db->join('tutorschedules ts', 'ts.tutorScheduleID = tses.tutorScheduleID');
$this->db->join('tutorexpertise te', 'ts.tutorID = te.tutorID');
$this->db->join('tutors tu', 'ts.tutorID = tu.tutorID');
$this->db->join('subjects s', 's.subjectID = te.subjectID');
$this->db->where('s.subjectID', $subjectID);
$this->db->where('ts.tutorScheduleID', $tutorScheduleID);
$this->db->where('tses.status', 'Approved');
return $this->db->get()->result_array();
}

function view_pending_sessions()
{
    $this->db->select('t.*, ut.userID uteeuid, ut.lastName uteeLN, ut.firstName uteeFN, ut.username
uteeUN, ut.emailAddress uteeEmail, ur.lastName urLN, ur.firstName urFN, s.subjectCode, tsr.dayofweek
tsrdow, tbr.timeStart tbrTS, tbr.timeEnd tbrTE');

    $this->db->join('tutees tees', 't.tuteeID = tees.tuteeID');
    $this->db->join('users ut', 'ut.userID = tees.userID');
    $this->db->from('tutorialsessions t');
    $this->db->join('subjects s', 't.subjectID = s.subjectID');
    $this->db->join('tutors tr', 'tr.tutorID = t.previousTutorID', 'left');
    $this->db->join('tutors ta', 'ta.tutorID = t.tutorID', 'left');
    $this->db->join('users ur', 'ur.userID = tr.userID', 'left');
    $this->db->join('users ua', 'ua.userID = ta.userID', 'left');
    $this->db->join('tutorschedules tsr', 'tsr.tutorScheduleID = t.tutorScheduleID', 'left');
    $this->db->join('timeblocks tbr', 'tbr.timeblockID = tsr.timeblockID', 'left');
    $this->db->or_where('t.status', 'Pending');
    $this->db->or_where('t.status', 'Cancel Pending');
    $this->db->or_where('t.status LIKE', 'Change Pending%');
    $this->db->order_by('tutorialNo', 'asc');
    if(isset($params) && !empty($params))

```

```

    {
        $this->db->limit($params['limit'], $params['offset']);
    }

    return $this->db->get()->result_array();
}

/*
 * Get all tutorialsessions
 */

function view_for_tutors()
{
    $this->db->select('t.*, ut.userID uteeuid, ut.lastName uteeLN, ut.firstName uteeFN, ut.username
uteeUN, ur.lastName urLN, ur.firstName urFN, s.subjectCode, tsr.dayofweek tsrdow, tbr.timeStart tbrTS,
tbr.timeEnd tbrTE, ua.emailAddress');

    $this->db->from('tutorialsessions t');

    $this->db->join('tutees tees', 't.tuteeID = tees.tuteeID');

    $this->db->join('users ut', 'ut.userID = tees.userID');

    $this->db->join('subjects s', 't.subjectID = s.subjectID');

    $this->db->join('tutors tr', 'tr.tutorID = t.previousTutorID', 'left');

    $this->db->join('tutors ta', 'ta.tutorID = t.tutorID');

    $this->db->join('users ur', 'ur.userID = tr.userID', 'left');

    $this->db->join('users ua', 'ua.userID = ta.userID');

    $this->db->join('tutorschedules tsr', 'tsr.tutorScheduleID = t.tutorScheduleID');

    $this->db->join('timeblocks tbr', 'tbr.timeblockID = tsr.timeblockID');

    $this->db->where('ua.userID', $_SESSION['userID']);

    $this->db->order_by('tutorialNo', 'asc');

    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
}

```

```

    }
    return $this->db->get()->result_array();
}

/*
 * Get all tutorialsessions count
 */
function tutor_tutorialsessions_count()
{
    $this->db->from('tutorialsessions t');
    $this->db->join('tutors tr', 'tr.tutorID = t.previousTutorID', 'left');
    $this->db->join('tutors ta', 'ta.tutorID = t.tutorID', 'left');
    $this->db->join('users ua', 'ua.userID = ta.tutorID');
    $this->db->where('ua.userID', $_SESSION['userID']);
    return $this->db->count_all_results();
}
}

```

//Tutorschedule_model.php

<?php

class Tutorschedule_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get tutorschedule by tutorScheduleID

*/

function get_tutorschedule(\$tutorScheduleID)

{

return \$this->db->get_where('tutorschedules',array('tutorScheduleID'=>\$tutorScheduleID))->row_array();

}

/*

* Get tutorschedule where params

*/

function get_tutorschedule_where(\$params)

{

return \$this->db->get_where('tutorschedules', \$params)->row_array();

}

/*

* Get tutorschedule by tutorID

*/

```

function get_tutorsched($tutorID,$term,$sy)
{
    return $this->db->
>get_where('tutorschedules',array('tutorID'=>$tutorID,'term'=>$term,'schoolYr'=>$sy))->row_array();
}

```

```

/*

```

```

    * Get all tutorschedules count

```

```

*/

```

```

function get_all_tutorschedules_count()

```

```

{

```

```

    $this->db->from('tutorschedules');

```

```

    return $this->db->count_all_results();

```

```

}

```

```

//get all tutorschedule count with specific timeblock

```

```

function get_count_schedule_timeblock($timeblockID,$term,$sy)

```

```

{

```

```

    $this->db->from('tutorschedules');

```

```

    $this->db->where('timeblockID', $timeblockID);

```

```

    $this->db->where('term', $term);

```

```

    $this->db->where('schoolYr', $sy);

```

```

    return $this->db->count_all_results();

```

```

}

```

```

/*

```

```

    * Get all tutorschedules

```

```

*/

```

```

function get_all_tutorschedules($params = array())

```

```

{
    $this->db->order_by('tutorScheduleID', 'asc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    return $this->db->get('tutorschedules')->result_array();
}

/*
 * Get term tutorschedules
 */
function get_term_tutorschedules($params = array())
{
    $this->db->join('tutors t', 't.tutorID = ts.tutorID', 'left');
    $this->db->join('users u', 'u.userID = t.userID');
    $this->db->join('faculty fac', 'u.userID = fac.userID', 'left');
    $this->db->join('students stu', 'u.userID = stu.userID', 'left');
    $this->db->join('timeblocks tb', 'tb.timeblockID = ts.timeblockID');

    $this->db->order_by('tutorScheduleID', 'asc');
    if(isset($params) && !empty($params))
    {
        $this->db->limit($params['limit'], $params['offset']);
    }
    return $this->db->get('tutorschedules ts')->result_array();
}

/*

```

```

    * function to add new tutorschedule
    */
function add_tutorschedule($params)
{
    $this->db->insert('tutorschedules',$params);
    return $this->db->insert_id();
}

/*
    * function to update tutorschedule
    */
function update_tutorschedule($tutorScheduleID,$params)
{
    $this->db->where('tutorScheduleID',$tutorScheduleID);
    return $this->db->update('tutorschedules',$params);
}

}

```


//User_model.php

<?php

class User_model extends CI_Model

{

function __construct()

{

parent::__construct();

}

/*

* Get user by userID

*/

function get_user(\$userID)

{

return \$this->db->get_where('users',array('userID'=>\$userID))->row_array();

}

// Get user by username

function get_userID(\$username)

{

return \$this->db->get_where('users', array('username'=>\$username))->row_array();

}

/*

* Get all users count

*/

function get_all_users_count()

{

```

        $this->db->from('users');

        return $this->db->count_all_results();
    }

    /**
     * Get all users
     */
    function get_all_users($params = array())
    {
        $this->db->select('u.*, stu.studentNo, pr.programCode');
        $this->db->join('students stu', 'stu.userID = u.userID');
        $this->db->join('programs pr', 'pr.programID = stu.programID');
        $this->db->order_by('userID', 'asc');
        if(isset($params) && !empty($params))
        {
            $this->db->limit($params['limit'], $params['offset']);
        }
        return $this->db->get('users u')->result_array();
    }

    /**
     * function to add new user
     */
    function add_user($params)
    {
        $this->db->insert('users', $params);
        return $this->db->insert_id();
    }

```

```
/*
* function to update user
*/
function update_user($userID,$params)
{
    $this->db->where('userID',$userID);
    return $this->db->update('users',$params);
}

/*
* function to delete user
*/
function delete_user($userID)
{
    return $this->db->delete('users',array('userID'=>$userID));
}
}
```


