

questions.view.php

```
<?php /* @ */
class Questions extends ViewPage
{
    var $testid = 0;
    var $question_count = 0;
    var $correct_count = 0;
    var $testname = "";
    var $send = false;
    var $subm = false;
    var $q_index = 1;
    var $q_index_post = 1;

    var $dbTempRow;

    function Questions()
    {
        $this->ViewPage();
        $this->onLoad();
    }

    function onLoad()
    {
        //print(session_id());
        $this->preLoad();
        $this->doAction();
        $this->buildPage();
    }

    function preLoad()
    {
        $row_db = $this->getFromDB();
        if($row_db)
        {
            $this->dbTempRow = $row_db;
            $this->send = true;
        }

        $testid = 0;
        if($this->send)
        {
            $this->q_index_post = $this->utils->Post("q_index");

            $testid = $this->dbTempRow["testid"];
            $this->testname = $this->dbTempRow["testname"];
            $this->question_count = $this->dbTempRow["q_count"];
            $this->q_index = $this->dbTempRow["q_index"];
            $this->correct_count = $this->dbTempRow["correct_count"];
        }
    }
}
```

questions.view.php

```
    }
    else
    {

        $testid = $this->utils->Get("testid");

        //question info begin
        $table = $this->db->prefix."tests";
        $where = array(
            'testid' => array("equal",
                $testid)
        );
        $sql = $this->sql->getSelectStatement($table, false, $where);

        $result = $this->db->query($sql);
        if($row = $this->db->fetch($result))
        {
            $this->testname = $row['name'];
            $this->question_count = $row['question_count'];
        }
        //question info end
    }

    if(is_numeric($testid) && $testid != 0)
    {
        $this->testid = $testid;
    }
}

#region actions

function doAction()
{
    $id = $this->utils->Post('action_id');

    $action = $this->utils->Post('action');

    switch($action)
    {
        case "submit":
            $this->submit();
            //$this->send = true;
            break;
    }
}

function submit()
{
```

questions.view.php

```
$testusr = $this->utils->GetFromConfig("testusr");
$hidden_q_ids = $this->dbTempRow["q_ids"];
$corrAnsStr = $this->dbTempRow["correct_ans"];
$corrAns = $this->parseCorrAns($corrAnsStr);

$q_ids = $this->parseQIDSString($hidden_q_ids);

$index = $this->q_index-1;
$ans = $_POST["ans"];

/*echo "index".$index."<br/>
";
echo "DB index".$tempdata_row['q_index']."<br/>
";
echo "ans".$ans."<br/>
";
echo "q_ids".$q_ids."<br/>
";
echo "corrAns".$corrAns."<br/>
";
echo "hidden_q_ids".$hidden_q_ids."<br/>
";
echo "testusr".$testusr."<br/>
";
*/

$qid = $q_ids[$index][0];
if($q_ids[$index][1] == "1")
{
    if(count($ans) == count($corrAns[$qid]))
    {
        $corr = true;
        for($j=0; $j<count($ans);$j++)
        {
            if(!in_array($ans[$j],$corrAns[$qid]))
                $corr = false;
        }
        if($corr)
        {
            $this->correct_count++;
            $check_sql = "SELECT * FROM
".$this->db->prefix."testuser WHERE
                (memberid = '".$_SESSION["UID_m"]."')
                AND (testid = '".$this->testid."')
                AND (log_id = 0)
                AND (sessionid = '".session_id()."')
";

```

questions.view.php

```
        if ($testusr > $this->db->num_rows($check_sql))
        {
            $stst_columns = array(
                'memberid' => $_SESSION["UID_m"],
                'testid' => $this->testid,
                'question_info' => $qid,
                'sessionid' => session_id()
            );
            $stst_sql =
$this->sql->getInsertStatement($this->db->prefix."testuser", $stst_columns);
            $this->db->query($stst_sql);
        }
    }
}
else
{
    if($ans == $corrAns[$qid][0])
    {
        $this->correct_count++;
        $check_sql = "SELECT * FROM ".$this->db->prefix."testuser
WHERE
                (memberid = '".$_SESSION["UID_m"]."')
                AND (testid = '".$this->testid."')
                AND (log_id = 0)
                AND (sessionid = '".session_id()."')
";
        if ($testusr > $this->db->num_rows($check_sql))
        {
            $stst_columns = array(
                'memberid' => $_SESSION["UID_m"],
                'testid' => $this->testid,
                'question_info' => $qid,
                'sessionid' => session_id()
            );
            $stst_sql =
$this->sql->getInsertStatement($this->db->prefix."testuser", $stst_columns);
            $this->db->query($stst_sql);
        }
    }
}

if($this->q_index == $this->question_count)
{
    $memberid = $_SESSION["UID_m"];
    $stable = $this->db->prefix."log";
```

questions.view.php

```

        $percent = $this->correct_count * ( 100 / $this->question_count
    );
    $correct_percent =
$this->utils->GetFromConfig("correct_percent");
    $cert_exp = $this->utils->GetFromConfig("cert_exp");
    $nomination = "none"; $nom_have = false;
    foreach ($correct_percent as $percent_name => $value)
    {
        if ($percent>=$value) { $nomination = $percent_name;
$nom_have = true; }
    }
    $correct_percent_lang = $this->fromLangIndex('correct_percent');
    $skr_txt = "";
    if ($nom_have)
    {
        /* B: Testi kechib */
        $nom_have_txt = $this->fromLang('correct_test');
        $skr_txt = $this->db->flread("question_final");
        /* E: Testi kechib */
    } else {
        /* B: Testi keche bilmeyib */
        $nom_have_txt = $this->fromLang('incorrect_test');
        $skr_txt = $this->db->flread("question_not");
        /* E: Testi keche bilmeyib */
    }

    $check_sql = "SELECT *
        FROM ".$this->db->prefix."log L
        WHERE
            (L.memberid = '".$memberid.'"
            AND (L.testid = '".$this->testid.'"
            AND (L.total = '".$this->question_count.'"
            AND (L.correct = '".$this->correct_count.'"
            AND (L.percent = '".$percent.'"
            AND (L.nomination = '".$nomination.'"
            AND (L.ordered = '0')
            ";
    if ($this->db->num_rows($check_sql)==0)
    {
        /* begin: if haven't any ID */
        $find_id = $this->db->find_id($table);
        $columns = array(
            'memberid' => $memberid,
            'testid' => $this->testid,
            'total' => $this->question_count,
            'correct' => $this->correct_count,
            'created' => time(),
            'percent' => $percent,

```

questions.view.php

```
        'nomination' => $nomination,
        'exp_time' => (time()+$cert_exp*31536000)
    );
    $sql = $this->sql->getInsertStatement($table, $columns);

    $this->db->query($sql);

    $columns = array(
        'log_id' => $find_id
    );

    $where = array(
        'memberid' => array("equal", $_SESSION["UID_m"]),
        'testid' => array("equal", $this->testid),
        'log_id' => array("equal", 0),
        'sessionid' => array("equal", session_id())
    );

    $sql =
    $this->sql->getUpdateStatement($this->db->prefix."testuser", $columns,
    $where);
    $this->db->query($sql);

    /* B: Send mail */
    $member_sql = "SELECT * FROM ".$this->db->prefix."members M
    WHERE M.userid = '".$_SESSION["UID_m"]."'";
    $member_result = $this->db->query($member_sql);
    if($member_row = $this->db->fetch($member_result))
    {
        if ( (strlen($member_row['email'])>7) AND ($nomination !=
    'none' ) )
        {
            $headers = "MIME-Version: 1.0\r \n";
            $headers .= "Content-type: text/html; charset=UTF-8\r
    \n";

            $headers .= "From: \"Ilk Addimlar\"
    <cavabyazma@firststeps.az>\r \n";
            $headers .= "X-Priority: 3\r \n";
            $headers .= "X-MSMail-Priority: High\r \n";
            $headers .= "X-Mailer: Just My Server";
            $Pother = "
                <b>İmtahanın adı</b>: ".$this->testname."<br/><br/>
                <b>Düzgün cavablar</b>:
                ".$this->correct_count."/". $this->question_count."<br/><br/>
                <b>Nominasiyanız</b>:
                ".$correct_percent_lang[$nomination]."
                ".$this->fromLangIndex('medal')."<br/><br/>

```

questions.view.php

```
        ".$sekr_txt;
        mail($member_row['email'], "Imtahanin neticesi",
$Pother, $headers);
    }
}
/* E: Send mail */

/* end: if haven't any ID */
}

$this->subm = true;
$correct_txt = ""; $incorrect_txt = "";
if ($percent>0) $correct_txt = $percent."%";
if ((100-$percent)>0) $incorrect_txt = (100-$percent)."%";
$this->template->assign_block_vars('percents', array(
    'TESTNAME' => $this->testname,
    'CORRECT' => $percent,
    'CORRECT_COUNT' => $this->correct_count,
    'CORRECT_TXT' => $correct_txt,
    'INCORRECT' => (100-$percent),
    'INCORRECT_TXT' => $incorrect_txt,
    'INCORRECT_COUNT' =>
($this->question_count-$this->correct_count),
    'TOTAL_COUNT' => $this->question_count,
    'NOMINATION' => $correct_percent_lang[$nomination].
".$this->fromLangIndex('medal'),
    'STATE' => $nom_have_txt,
    'MSG' => $this->fromLang('final'),
    'EKR_TXT' => $sekr_txt
));
}

$q_qid = $this->utils->Post("qid");
$qidArr = $this->getElementArray($q_ids, 0);
$qid_index = array_search($q_qid, $qidArr);
$this->q_index = $qid_index + 2;
//print($qid_index."_a");
//$this->q_index++;

$this->q_index_post++;
}

function
saveToDB($testid, $q_ids, $q_index, $testname, $q_count, $correct_ans, $correct_cou
nt)
{
```

questions.view.php

```
        $table = $this->db->prefix."tempdata";

        $where = array(
            'sessionid' => array("equal",
session_id())
        );
        $sql = $this->sql->getDeleteStatement($table, $where);
        $this->db->query($sql);

        $columns = array(
            'testid' => $testid,
            'q_ids' => $q_ids,
            'q_index' => $q_index,
            'testname' => $testname,
            'q_count' => $q_count,
            'correct_ans' => $correct_ans,
            'correct_count' => $correct_count,
            'sessionid' => session_id()
        );
        $sql = $this->sql->getInsertStatement($table, $columns);

        $this->db->query($sql);
    }

    function getFromDB()
    {
        $table = $this->db->prefix."tempdata";
        $where = array(
            'sessionid' => array("equal",
session_id())
        );

        $sql = $this->sql->getSelectStatement($table, false, $where);

        $result = $this->db->query($sql);

        $row = null;
        if($row = $this->db->fetch($result))
        {
            print("");
        }

        return $row;
    }

    function getCorrectAnswers($q_ids_id)
    {
        $retArr = array();
```

questions.view.php

```
$q_ids_in = implode(",", $q_ids_id);
if($q_ids_in == "")
    $q_ids_in = "0";

$sql = $this->sql->selectCorrectAns;
$sql = str_replace("{PREFIX}", $this->db->prefix, $sql);
$sql = str_replace("{0}", $q_ids_in, $sql);

$result = $this->db->query($sql);

while($row = $this->db->fetch($result))
{
    $id = $row['questionid'];
    $a_id = $row['answerid'];

    $retArr[$id][] = $a_id;
}

return $retArr;
}
#endregion

#region page building

function buildPage()
{
    $this->buildMain();

    if($this->subm)
    {
        $this->template->assign_block_vars('mess', array());
    }
    else
    {
        $this->buildList();
    }
}

function buildMain()
{
    $this->template->assign_vars(array(
        'TITLE' => $this->fromLang('title'),
        'SUBMIT' => $this->fromLang('submit'),
        'TEST' => $this->fromLang('test'),
        'REQUEST_ERROR' => $this->fromLang('form_request'),
        'TIME' => $this->fromLang('time'),
        'MESSAGE' => $this->fromLang('send').$this->fromLang('final'),
        'TEST_TIME' => $this->utils->GetFromConfig("test_time"),
```

questions.view.php

```
        'LANG' => $this->lang,
        'CANCEL_TEST' =>
$this->fromLang('cancel_test')
    ));
}

function buildList()
{
    $this->buildListPage();
}

function buildListPage()
{
    $hidden_q_ids = "";
    $corrAnsStr = "";

    if($this->send)
    {
        $hidden_q_ids = $this->dbTempRow["q_ids"];
        $corrAnsStr = $this->dbTempRow["correct_ans"];
    }
    else
    {
        $ques = $this->getQuestionsIds($this->testid);
        $ques = $this->getRandomElements($ques, $this->question_count);

        for($i=0; $i<count($ques); $i++)
        {
            $sid = $ques[$i][0];
            $multiple = $ques[$i][1];

            if($i != 0)
                $hidden_q_ids .= ",";
            $hidden_q_ids .= $sid."_".$multiple;
        }
    }

    $qidArr = $this->parseQIDSString($hidden_q_ids);
    if(!$this->send)
    {
        $qids_str = $this->getElementArray($qidArr, 0);
        // $qids_str = implode(",", $qids_str);
        $arrCorrAns = $this->getCorrectAnswers($qids_str);
        $corrAnsStr = $this->getStrCorrAns($arrCorrAns);
    }
}
```

questions.view.php

```
    $ques_index_text = $this->fromLang('question_index');
    $ques_index_text = str_replace("{TOTAL}", $this->question_count,
$ques_index_text);
    $ques_index_text = str_replace("{CURR}", $this->q_index,
$ques_index_text);

    $qid_mul
=$qidArr[$this->q_index-1][0]."_".$qidArr[$this->q_index-1][1];
    $this->template->assign_vars(array(
        //'TESTID' => $this->testid,
        'TESTNAME' => $this->testname,
        //'Q_COUNT' => $this->question_count,
        //'HIDDEN_Q_IDS' => $hidden_q_ids,
        //'Q_INDEX' => $this->q_index,
        'QUESTION_INDEX_TEXT' => $ques_index_text,
        'QID_MUL' => $qid_mul
        //'CORRECT_ANS' => $corrAnsStr,
        //'CORRECT_COUNT' => $this->correct_count
    ));

    $this->saveToDB($this->testid, $hidden_q_ids, $this->q_index,
$this->testname, $this->question_count, $corrAnsStr, $this->correct_count);

    //section 2 begin

    $alphabet = "abcdefghijklmnopqrstuvwxyz";

    $qid = $qidArr[$this->q_index-1][0];

    $table = $this->db->prefix."questions";
    $where = array(
        'questionid' => array("equal",
$qid)
    );
    $sql = $this->sql->getSelectStatement($table, false, $where);

    $result = $this->db->query($sql);

    if($row = $this->db->fetch($result))
    {
        $id = $row['questionid'];
        $question = $row['question'];
        $multiple = $row['multiple'];

        $this->template->assign_block_vars('list', array(
            'QID' => $id,
```

questions.view.php

```
        'QUESTION' => $question
    ));

    //answers begin
    $answers = $this->getAnswers($id);
    $answers = $this->getRandomElements($answers, 0, true);

    for($j=0; $j<count($answers); $j++)
    {
        $id_a = $answers[$j]['id'];
        $answer = $answers[$j]['answer'];
        $isright = $answers[$j]['isright'];

        // $index_a = $j+1;
        $index_a = $alphabet[$j];
        $this->template->assign_block_vars('list.answers', array(
            'ANSWER' => $answer,
            'ID' => $id_a,
            'INDEX' => $index_a
        ));

        if($multiple == "1")

$this->template->assign_block_vars('list.answers.multiple', array());
        else

$this->template->assign_block_vars('list.answers.nomultiple', array());
    }
    //answers end
}
}

function getAnswers($qid)
{
    $retArr = array();

    $table = $this->db->prefix."answers";
    $where = array(
        'questionid' => array("equal",
        $qid)
    );
    $sql = $this->sql->getSelectStatement($table, false, $where);

    $result = $this->db->query($sql);

    $i = 0;
    while($row = $this->db->fetch($result))
    {
```

questions.view.php

```

        $id = $row['answerid'];
        $answer = $row['answer'];
        $isright = $row['isright'];

        $retArr[$i] = array('id' => $id,
                           'answer' => $answer,
                           'isright' => $isright,
                           );

        $i++;
    }

    return $retArr;
}

function getStrCorrAns($arrCorrAns)
{
    $str = "";
    if(count($arrCorrAns)<1)
        return $str;

    $i=1;
    foreach ($arrCorrAns as $key => $value)
    {
        if($i!=1)
            $str .= ",";

        $str .= $key."_";
        for($j=0; $j<count($value); $j++)
        {
            if($j!=0)
                $str .= ":";
            $str.=$value[$j];
        }

        $i++;
    }

    return $str;
}

function parseCorrAns($corrAnsStr)
{
    $retArr = array();

    $q = explode(",", $corrAnsStr);

    for($i=0; $i<count($q); $i++)

```

questions.view.php

```
{
    $arr = split("_", $q[$i]);
    $id = $arr[0];

    $ans = explode(":", $arr[1]);
    for($j=0; $j<count($ans); $j++)
    {
        $retArr[$id][] = $ans[$j];
    }
}

return $retArr;
}

function parseQIDSString($q_ids_str)
{
    $retArr = array();

    $q_ids = explode(",", $q_ids_str);

    for($i=0; $i<count($q_ids); $i++)
    {
        $arr = split("_", $q_ids[$i]);
        $retArr[$i] = $arr;
    }

    return $retArr;
}

function getElementArray($sourceArr, $index)
{
    $retArr = array();

    for($i=0; $i<count($sourceArr); $i++)
    {
        $retArr[$i] = $sourceArr[$i][$index];
    }

    return $retArr;
}

function getQuestionsIds($testid)
{
    $retArr = array();
    /*
    $table = $this->db->prefix."questions";
    $where = array(
        'testid' => array("equal",
```

questions.view.php

```
$testid)
    );
    $sql = $this->sql->getSelectStatement($table, false, $where);
    */

    $sql = $this->sql->selectQuestions;
    $sql = str_replace("{PREFIX}", $this->db->prefix, $sql);
    $sql = str_replace("{0}", $testid, $sql);

    $result = $this->db->query($sql);

    $i = 0;
    while($row = $this->db->fetch($result))
    {
        $id = $row['questionid'];
        $multiple = $row['multiple'];

        $retArr[$i] = array($id, $multiple);
        //$retArr[$i] = $id;
        $i++;
    }

    return $retArr;
}

function getRandomElements($sourceArr, $count, $rand_sort = false)
{
    $sCount = count($sourceArr);
    if($sCount <= $count && !$rand_sort)
        return $sourceArr;

    if($rand_sort)
        $count = $sCount;

    $retArr = array();

    while(count($retArr) < $count)
    {
        $index = rand(0, $sCount-1);

        if(!in_array($sourceArr[$index], $retArr))
        {
            $retArr[count($retArr)] = $sourceArr[$index];
        }
    }

    return $retArr;
}
```

questions.view.php

```
    }  
  
    #endregion  
}  
?>  
<?  
$questions = new Questions();  
$questions->template->pparse('questions');  
?>
```