Proceedings of ICCC 2004, Băile Felix Spa-Oradea, Romania
pp. 346-354, 2004

# Romanian ICT Education and Training Institutions Platform

Eugen Petac, Lucian Novac

**Abstract:** This application allows the addition of new information and wide documentation from persons and companies that seek to take advantage of professional ICT (Information and Communications Technology) services. The development of this project is a step in assuring an optimal environment for free information exchanging development in the wide field of ICT professional training. The application works on a Linux system with an Apache Web Server compiled with PHP and MySQL Server. Key Words: Information Society; ICT Education; ICT Training; Web Application.

## 1 Introduction

The term *Information Society* describes an economy and a society in which the access, acquisition, storage, processing, transmission, spreading and using knowledge and information plays a decisive role [1], [9]. The Information Society is made of five layers: the users layer, the applications layer, the informational infrastructure layer, the institutional layer and the legislative layer [9]. The Romanian Classification of Activities in the National Economy Ű CAEN ensures [3] the identification of all activities and their encoding in a unitary system. This allows the organization, rationalizing and information of the social-economical informational fluxes, creating the processing facilities for the integration in the international and national systems of presentation and analysis of information. The professional forming programs ensure the gaining of professional qualifications according to the nation-wide acknowledged occupational standards approved by the Romanian Council for Occupational Standards and Certification Ű COSA. This is the Romanian organism for certification of professional qualifications ensures the quality of the system by authorizing the evaluation centers, by monitoring their activity, by evaluating and certification of evaluators. EUQuaSIT Ű European Qualification Strategies in Information and Communications Technology (www.euquasit.net) is a transnational project (funded by the European Commission, Leonardo da Vinci II project, 2001-2004) being carried out since 2001 involving partners of five European countries: Czech Republic, Germany, Netherlands, Portugal, Romania. The project analyzed the specific demands of companies within their ICT workforce and to what extend different vocational training strategies in partner countries fulfill their needs.ă

**Romanian ICT Education and Training Institutions Platform** is a portal application (Romanian project, part of EUQuaSIT [11]) for dynamic registration and analysis of information concerning Institutions in Romania which offer education and training, of all kinds, in ICT field Ű Information and Communications Technology. This application allows the addition of new information and wide documentation from persons and companies that seek to take advantage of professional ICT services. The development of this project is a step in assuring an optimal environment for free information exchanging development in the wide field of ICT professional training. ICT Education and Training levels, classified according to SEDOC Ű training degrees assigned by the European Community [2], are listed in Description section [7] with qualification examples corresponding to all levels: 2, 3, 4, 5B and 5M.

## 2 Platform Design

The application works on a Linux system with an Apache Web Server compiled with PHP [13] and MySQL Server. Requirements: PC Server; Linux; Sendmail; Apache2.0.48 [10]; PHP4.3.4 [13]; MySQL4.0.17 [12]; PhpMyAdmin2.5.5-pl1 [14] (optional). In Figure 1 we present the Web application environment diagram. The mail server (Sendmail) is being used for sending an e-mail to the administrator of the **database** and one to the user, for registration confirmation. Installation of the PHP language has been installed with the following configuration statement: *"./configure Űwith-mysql –with-apxs2=/usr/local/apache2/ bin/apxs"*. Have occurred the following configuration changes in the file php.ini: `"upload_max_filesize=2M","upload_tmp_dir=`

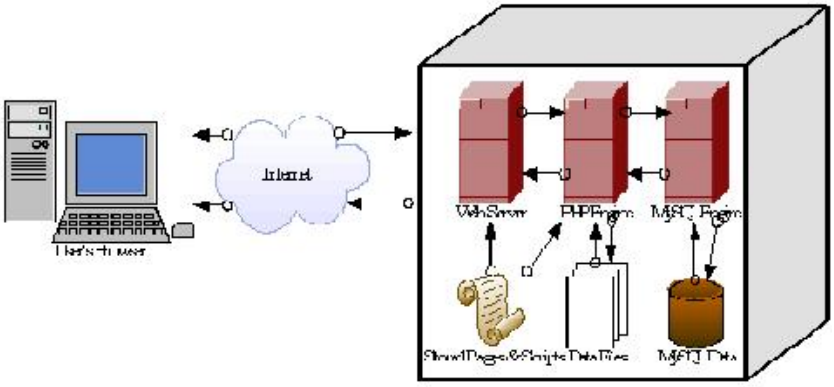

Figure 1: The Web application environment diagram

/tmp","mysql.default_socket=/var/lib/mysql/mysql.sock". This site has been projected to be most modular as possible. The purpose of this modularization was not to be written code in dozens of files, but to exist a Ďtemplate" which cumulates all the code. Applying existence conditions to a directory it could make the difference between: languages (English and Romanian), ICT levels (5M, 5B, 4, 3 and 2). So, code has been written only in a few files: *afisare.php, plan.php, afisareplan.php, adaug1.php, ..., adaug4.php, constructie.php, trimite.php, login.php and included files: functii.inc, head.inc, initsesid.inc and refacsesid.inc, and the original path to these "template" files is "/en"*.

The rest of the files and paths are links to these original files and path. So we present in Figure 2 a file and path link diagram of the application. We will sample the structure of the database for level 5M, tables Facultati and PlanInvatamant.

CREATE TABLE **'Facultati'** *('F' bigint(6) unsigned NOT NULL auto_increment, 'Institutie' varchar(120) default NULL, 'Judet' varchar(30) default NULL, 'Facultate' varchar(100) default NULL, 'Specializare' varchar(200) default NULL, 'Forma' varchar(10) default NULL, 'PerioadaStudii' float unsigned default '0', 'Adresa' varchar(100) default NULL, 'Link' varchar(150) default NULL, 'NrStudenti' bigint(5) unsigned default '0', PRIMARY KEY ('F'), UNIQUE KEY 'F' ('F'), KEY 'F_2' ('F')) TYPE=MyISAM;*

CREATE TABLE **'PlanInvatamant'** *('P' bigint(6) unsigned NOT NULL auto_increment, 'PSpecializare' bigint(6) unsigned default NULL, 'An' tinyint(3) unsigned default NULL, 'Semestru' tinyint(3) unsigned default NULL, 'Forma' varchar(5) default NULL, 'Plan' text, 'Link' varchar(100) NOT NULL default '', PRIMARY KEY ('P'), UNIQUE KEY 'P' ('P'), KEY 'P_2' ('P')) TYPE=MyISAM;*

We currently use PhpMyAdmin, a tool to manage the projects' database (Figure 3), written in PHP and intended to handle the administration of MySQL over the Web. We used it to create and drop the database, create/drop/alter tables, delete/edit/add fields, execute SQL statements, manage keys on fields, manage privileges, export and import data into and from MS Access, MS Excel.

## 3 Security advices

### 3.1 Global variables:

Having global variables ON we could send to forms easier:

```
$variable instead of $_GET['variable'],
```

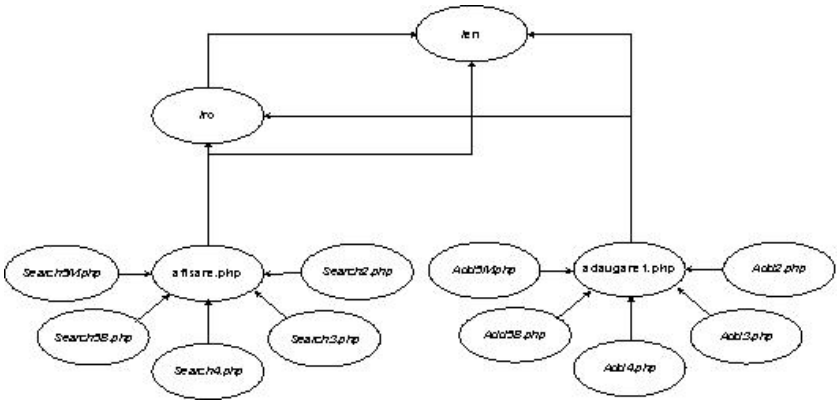


Figure 2: File and path link diagram

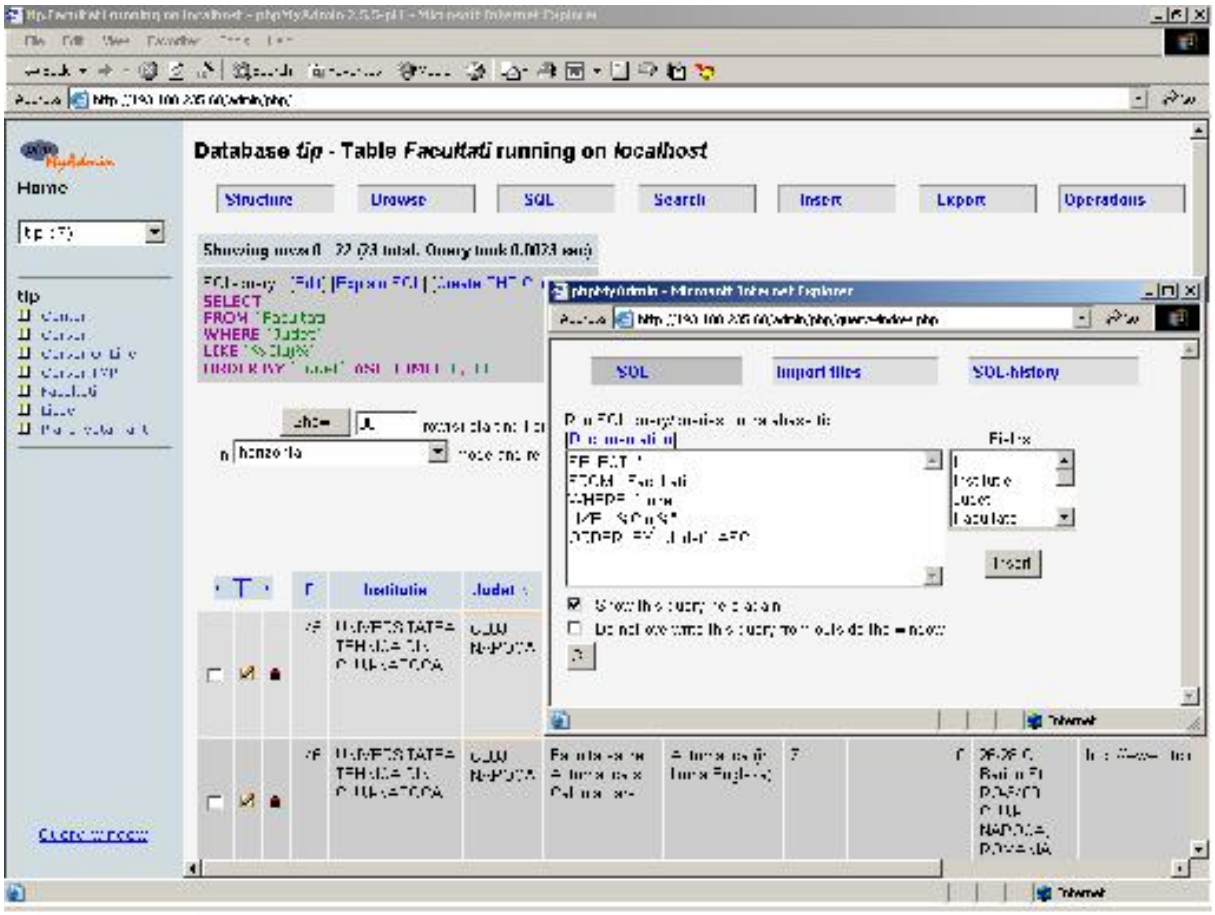


Figure 3: Manage the project database with PhpMyAdmin

same as for POST or FILES. If the script is not carefully managed, global variables can become a high level security risk [15]. That's why php.ini is distributed with globals=off in the last versions.

### 3.2 "Magic" quotation marks:

If in *php.ini magic_quotes_gpc=OFF* it should be used the function addslashes to precede quotation marks from data sent by users, with "\"character. *If magic_quotes_gpc=ON*, PHP automatically adds "\"character before quotation marks, if they're *OFF*, the quotes from input can cause serious security problems. For example, let us suppose that username and password checking SQL syntax for a login page is

```
SELECT * FROM users WHERE name='$name' and password='$password'
```

is successfully executed and returns a result (actually, returns all the registrations from the database).

It should be checked if the SQL syntax returns a single row, not that the interrogation was successfully executed:

/* not like this */

```
$sql = "SELECT * FROM users WHERE name='$name' and password='$password'";
$resource = mysql_query($sql) ;
if($resource)
```

..- certification... /* like this */

```
$sql = "SELECT * FROM users WHERE name='$name' and password='$password' ";
$resource = mysql query($sql);
if(mysql num rows($resource) == 1)
```

... certification...

**Note:** *magic_quotes_gpc* are by default *ON* in *php.ini* to protect the application from such attacks, but it is recommended to check php.ini before the renounce to addslashes function.

### 3.3 Inclusion:

It's necessary [15], [16] to avoid files "visible" inclusion in the form *http://site.com/file.php?file =search.html* to include file to a page. Little absence of mind the attacker can access sensitive information from the system. It is not recommanded to include foreign files. PHP can "include" files found on other server if the setting URL fopen wrappers is activated in php.ini. In the above example, could access *http://site.com/filer.php?file=http://www.bad.com/ badscript.php* to include in the file a script located on another server and so to obtain access to all system resources to which PHP has access, letting him to execute system commands, display confidential information or erase the database. Setting *allow_url_fopen==0ff* in php.ini, won't allow inclusion of files from another server.

### 3.4 Forms:

POST method should be used in forms when the information is to be entered in database. Let us suppose that we have a form in which users can enter their opinion about this site, form which is processed by a file named *userinput.php:*

```
form.html:
<form action="userinput.php" method="GET">
```

```
<textarea name="my_opinion"
</textarea>
<input type="submit">
</form>

userinput.php:
<?
mysql_connect(...);
mysql_query("insert into opinions values \begin{verbatim} ('$myopinion')");?>
```

If global variables are OFF in php.ini or the transmission method of the form is GET, a bad intentioned user can access address

```
http://site.com/userinput.php?myopinion=database\%20has%20been%20hacked
```

and will enter the opinion "database has been hacked" in your database without entering actually in the site.

It's possible that the hacker writes a script which accesses address without stopping for some hours. If his Internet connection is good, it is possible that after a while to find out that the free space on the server has been lowered, and millions of opinions say the same thing: "database has been hacked".

Checking the origin of the requests to the server is very important also for login forms. Someone who knows the username can try (and with some luck to succeed) to find the password in such a way.

Having available a dictionary (exhaustive search), intuition or a password generator can easily and rapidly access the address:

```
www.site.com/login.php?username=You&password=aaa,
www.site.com/login.php?username=You&password=aab,
www.site.com/login.php?username=You&password=aac.
```

And so on until the discovery of the valid username and password combination. In this case it's necessary to put a supplementary protection that will not allow more that three consecutive failed login tries for a username.
This problem is rapidly resolved using sessions. When the user sends username and password to login.php, we can set a session variable $_ SESSION[Ślogin_ count'] which remembers the number of tries. When the value of this variable exceeds 3 (three failed tries) you will not even interrogate the database to check the forth try.
The variable $_ SESSION['login_ count'] will be available in memory as long as the browser is opened and the session is active (the default option of PHP maintaining sessions active is one hour). If the attacker awaits for an hour or closes his browser, then the session will close and can try again for three times to login.
Many times this security measure is sufficient [15], [16] to prevent tries of finding out the users passwords. Browser closing and opening for 3 times tries sufficient discouraging to determine a hacker to hack somewhere else.

### 3.5 Extensions

A usual practice is to give extension .inc to files that contain libraries of functions which will be included and used in the coded. PHP does not parse .inc extension files and these are directly called, are sent as plain text to the browser.
It is necessary to avoid exposure of information so .php extension should be given to the files, so that files will be executed instead of being displayed. It shouldn't be included sensitive information (like the name and password which permit connection to MySQL server) in .inc, .txt or .html extension files which can be accessed and displayed.
Another method we used to avoid exposure of information in .inc files was to deny the display of these files in the Apache2 configuration file httpd.conf, so we blocked the view in the include directory containing .inc files like this:

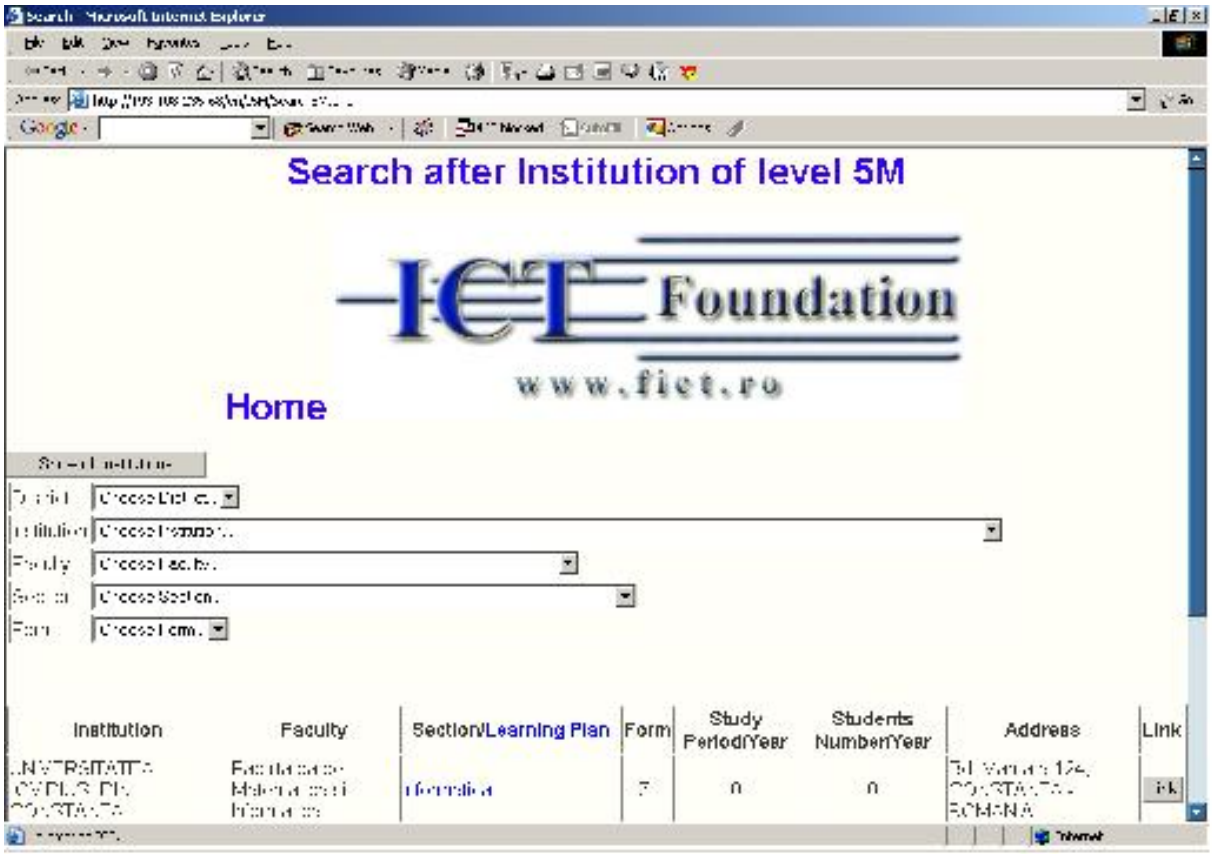


Figure 4: Search Form Interface, 5M Level

```
Alias /include/ "/usr/local/apache2/htdocs/include/"
<Directory "/usr/local/apache2/htdocs/include/">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Deny from all
</Directory>
```

These are only few of the methods we can use to secure the application. More information about security can be found in [15], [16].

## 4 Application Guide

### 4.1 Search Form:

The interface for Search Form, with Level 5M content, is displayed in Figure 3. We present below the content of files and their use mode.

**afisare.php**

First of all on this page there exists a url request of the page through you got here. So, if you opened "TIP_en.htm" file and click on a link of this page, will link you to a file in "/en" directory. For example, if you click on ĎSearch Institution ? Level Institution 5M", then the page which will open will be Ď/en/Search5M.php", displaying its

contents in English. If you're on the Romanian page, "TIP_ro.htm", then at a link opening a "/ro" path is opened and Romanian content will be displayed. Home page link, ĎHome" will take you back to ĎTIP_en.htm" or "TIP_ro.htm" depending on the language type displayed on the current page. There are 3 to 5 selection types, or searching criteria: after the district, institution, faculty(5M,5B), section or form. For each selection, the page will automatically refresh and below a table will appear with the following columns: Institution, Faculty, Section, Form, Address, and Link. On Link column appear buttons, which, being pushed will send you through a popup window to that section's web site. In case that the section has a learning plan it will be hyperlinked and if you click it, a new popup window will appear "afisareplan.php?F=x" where x is the index number of the section in the database.

**afisareplan.php**

This page displays a table with the learning plan split into 5 columns: year, semester, form, learning plan and a download button to download in *.doc* or *.pdf* format the document containing the section's learning plan.

**functii.inc** This file contains several functions, function connect(), which connects to the database, if cannot connect an error message will be displayed. Second, function schimba ($url,$tabel,$titlu,$ltoateinstitutiile,$ljudet,$lalegejudetul, $linstitutie,$lalegeinstitutia,$lfacultate,$lalegefaculatea , $lspecializare,$lalegespecializarea,$lforma,$lalegeforma,$lplandeinvatamant , $lperioadastudii,$lnumarstudenti,$ladresa,$sqlcat) , keeps the track with the path changes, from the $url parameter which used in a condition, for example, if (stristr($urlcurent,"/I5M" )) changes the following: table name, title of page and its header, in English or in Romanian, an SQL syntax $sqlcat , which concatenates to another and the name of the selection boxes $lxxxxxx.

### 4.2 Adding Form:

The group of files *"adaug*.php"*, where * stands for 1, 2, 3 and 4, is a group of institution registration forms, institution which has in its learning plan ICT disciplines.

**adaug1.php, adaug2.php, adaug3.php, adaug4.php, trimite.php, multumim.php** are files using sessions. We will take the case when a 5M level institutions section is added.

First page, *"adaug1.php"*, registers the following: institutions name, district, faculty, and section, form which has the following radio buttons values: **Zi, Zi/ID** or **Zi/ID/CTI**, other edit boxes like: study, address and link to the sections' site if it exists. All fields, marked by `"*"` , except "Link" are required.

When the user finishes completing all required fields "Next" button is to be pressed. If any required field is not completed, then an error message will be displayed. Second page, *"adaug2.php"*, registers learning plan of the section. This can be done by uploading a .doc or .pfd file by pressing "Browse..." button and then "Next", or can be done manually writing in each field the learning plan on each year, semester, and forma and then "Next".

Page *"adaug3.php"* uploads learning programs for each discipline: in "Discipline" field set the name of the discipline and next to upload the file specific to the discipline.

Pushing "Add Discipline" button a new discipline and its upload program file can be added. In the end "Next" button must be pushed for next step registration.

Forth page, *"adaug4.php"* registers users' certification information who registered current ICT institution. Information is to be completed in the following required fields: surname, name, profession, contact telephone, e-mail address and password, last two required for login.

In the end "Finish" button is to be pressed so that current registration to be finished. A new page "trimite.php" will be displayed with a next step confirmation message, and in the background will send an e-mail to the user a confirmation link. When the user confirm with a click on the hyperlinked confirmation address, then a thank you message will be displayed in the "multumim.php".

**multumim.php** This file contains a "Thank you message" and a "Login" button used for users' login without any validation and a "Home" button. Behind this display, an e-mail is sent to the users' e-mail with the following

data: user name (e-mail address) and password to login, which are used for data change or for new attachments. Also an e-mail will be sent to database administrator to warn him about a new registration in the database.

### 4.3 Login Form

"login.php" is a file used for login validation. The login form contains an e-mail and a password edit boxes and a "Login" button. If data is wrongfully entered error messages appear, or if the user tries 3 times is refused and thrown out from the login session for security purposes in connection with 3.4 chapter of this paper.

## 5 Conclusions

Part of European Project EUQuaSIT (www.euquasit.net) [2], [11], *Romanian ICT Education and Training Institutions Platform* is a portal application, for dynamic registration and analysis of information concerning Institutions in Romania which offer education and training, of all kinds, in ICT field. The application runs on Linux System using SendMail, Apache2.0.48, PHP4.3.4, MySQL4.0.17, PhpMyAdmin2.5.5-pl1, with all thabove free source packages [8]. PhpMyAdmin is a tool written in PHP intended to handle the administration of MySQL over the Web. Currently we used it to create and drop the database, create/drop/alter tables, delete/edit/add fields, execute SQL statements, manage keys on fields, manage privileges, export and import data into and from various formats (example MS Access, MS Excel). We currently use this tool to manage this projects' database. Each member can enter a national and European circuit, exchanging information between institutions which assure the educational process progress for ICT professional training and can enter a competitive offer-demand circuit, national and European, according to companies demands. The system is in use (http://www.fict.ro/TIP/ TIP_en.htm) and for further we intend to make a more complex search engine and more secure adding forms and to implement a complex system for data analyze.

## References

[1] ESIS II Report: Information Society Indicators in the CEEC countries, www.eu-esis.org/esis2proj/

[2] EUQuaSIT (Leonardo da Vinci II project Ű D/00/C/P/RF/91309), Interim Report, Flensburg University, July, 2002.

[3] HG 656/1987, Classification of Activities in the National Economy Ű CAEN, Romanian Official Monitor, Part I no. 301, 5 November 1997.

[4] E. Petac, D. Petac - An Analysis of ICT Policy and Strategies in Romania in European Context, Networking in Education and Research, Proceedings of RoEduNet International Conference, Iasi, Romania, June 5-6 2003, pp.167-174.

[5] Romanian Ministry of Education and Research, www.edu.ro.

[6] Romanian Ministry of Communications and Information Technology, Bucharest, Romania, www.mcti.ro.

[7] Romanian ICT Education and Training Platform, http://www.fict.ro/TIP/TIP_description.htm.

[8] L. Welling, L. Thomson Ű PHP and MySQL Web Development, Sams Publishing, USA, 2001.

[9] www.academiaromana.ro

[10] www.apache.com

[11] www.euquasit.net

[12] www.mysql.com

[13] www.php.net

[14] www.phpmyadmin.net

[15] www.secureреality.com.au

[16] www.securityfocus.com.

ICT Foundation, Romania
15 Timisana Street, Constanta - 900524, Romania
http://www.fict.ro
E-mail: office@fict.ro, novaclucian@yahoo.com