



**ПЛОВДИВСКИ УНИВЕРСИТЕТ “ПАИСИЙ
ХИЛЕНДАРСКИ”**

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

КАТЕДРА “КОМПЮТЪРНИ СИСТЕМИ”

ДИПЛОМНА РАБОТА

ТЕМА: “Електронен самоучител по РНР”

Дипломант:

Анна Любомирова Ламбова

Научен ръководител:

доц. д-р Коста Гъров

Факултетен номер: 0901182002

Специалност: Математика и информатика

Задочно обучение

Пловдив, 2011

СЪДЪРЖАНИЕ

Увод	3
I.ПРЕДМЕТ И ЗАДАЧИ НА ДИПЛОМНАТА РАБОТА	7
II. ПРОЕКТИРАНЕ НА ИНФОРМАЦИОННАТА СИСТЕМА.....	8
2.1 Организация на системата	8
2.2 Проектиране на базата от данни:	10
III. ПРОГРАМНА РЕАЛИЗАЦИЯ.....	14
3.1 Използвани технологии.....	14
3.1.1 HTML	14
3.1.2 CSS.....	15
3.1.3 SQL Език за сруктурирани запитвания.....	16
3.1.4 MySQL.....	17
3.1.5 PHP.....	18
3.1.6 JavaScript	24
3.1.7 Apache.....	25
3.2 Описание на реализираните на PHP страници съставлящи WEB базираната система	26
IV.Ръководство на потребителя	43
4.1 Системни изисквания	43
4.2 Описание на страниците	43
V.ЗАКЛЮЧЕНИЕ	54
ЛИТЕРАТУРА	55

УВОД

Развитието на информационните и комуникационните технологии (ИКТ) е неразделна част от живота на съвременното общество, а ползването и прилагането на ИКТ е от решаващо значение за постигане на икономически растеж на фирмите, които ги ползват.

През 2006г. практически всички предприятия са компютъризирани; 75 – 80% имат достъп до Интернет и 24% имат Интернет страници. В около 60% от предприятията компютрите се използват предимно в ежедневните бизнес дейности и счетоводството. Само 10% от предприятията използват информационни системи за управление, системи за автоматизация на производството или системи за управление на отношенията с клиентите. Едва 6,6% от предприятията използват електронни системи за планиране на ресурсите.

Споменаването на информационни технологии е неразривно свързано със споменаването на Интернет и всички услуги, които той предлага. Като най – популярната услуга е WWW (World Wide Web) услугата, чрез която се

предлага всякакъв вид информация на всеки, който се нуждае от нея, на всяко място по света.

Структурата на Интернет е базирана на технологията клиент – сървър. Това е ключово понятие за разбирането на функционирането на Световната мрежа. Принципа, на който действа тази технология е следния: клиента (това може да е браузър, FTP клиент или друга програма, в общия случай работещата на локалния компютър) се свързва към отдалечения компютър (наречен сървър), който доставя услугата заявена от клиента.

Основната част от данни в Интернет се пренася благодарение на два основни протокола. Това са IP и TCP. IP (Internet Protocol) е прост протокол, но е сърцето на Интернет. Той има основно две задачи: да осигурява IP адресите на подателя и получателя, както и да намери път през мрежата между подателя и получателя. TCP (Transport Control Protocol) служи за разделяне на информацията на пакети и транспортирането им. С негова помощ пакетите се събират отново при получателя.

В допълнение към протоколите TCP/IP, за комуникация в Интернет WEB използва и свой собствен протокол наречен, http (Hyper Text Transfer Protocol -протокол за трансфер на хипертекст), за обмени между браузърите и сървърите. HTTP се използва от браузърите за изискване на документи от

сървърите и от сървърите за изпращане на заявените документи към браузърите.

Цялата информация е организирана под формата на WEB сайтове, които представляват колекция от документи обединени от обща тема. За тяхното оформяне се използват специален език за програмиране, наречен HTML (Hyper Text Markup Language), или език за хипертекстови документи.

За реализация на WEB услугите разработчиците имат на разположение различни средства за разработване, като най – разпространените са: CGI, Java, ActiveX, ASP, PHP, Perl, Java Script.

А с навлизането на мобилни безжични технологии се появиха и WAP (Wireless Application Protocol) и WML(Wireless Markup Language).

WAP е протокол, стандарт за достъп до Интернет през безжични устройства. Той е отворена, глобална спецификация, позволяваща потребителите на мобилни средства лесен и бърз достъп до информация в интернет, независимо от разстоянията. Работи подобно на HTTP протокола, който е в основата на Интернет.

За комуникацията през WAP, се използва WML (Wireless Markup Language), език за форматиране, който е базиран на XML и така да се каже „братовчед” на HTML, тъй като също подобно на него е базиран на тагове. WML е част от WAP (Wireless Application Protocol). В такъв смисъл може да се

каже, че WML е за мобилния телефон това, което е HTML за брауъра на персоналния компютър.

Информацията най – често се съхранява в т. нар. „бази данни”, като в тях може да се съхранява информацията от многобройни източници в различни формати. За управлението на базите данни се използват специални програмни продукти - система за управление на базите данни, чрез нея се осигурява интерфейс на всички потребители с база от данни като им се предоставят езикови средства. Като най – често се използва SQL (Език за структурирани запитвания – Structures Query Language), който служи за създаване, модифициране, извличане и манипулиране на данни от базата данни.

Повечето уеб приложения за бази от данни постигат обединяването на WEB и базите от данни, чрез три слоя приложна логика. В основата ѝ се намира системата за управление на базите данни (DBMS) и база от данни.

На върха се намира клиентския брауър, използван като интерфейс на приложението. Между тях лежи по – голямата част от приложената логика, обикновено разработена със скриптов език от страна на сървъра, който може да си взаимодейства с DBMS и да декодира и произвежда код на HTML, използван за презентация в уеб брауъра на клиента.

ГЛАВА 1. ЦЕЛ И ЗАДАЧИ НА ДИПЛОМНАТА РАБОТА

Основна цел на дипломната работа е да създаде и развие система за самоулучение по РНР.

Проектът е свързан със създаване на възможност за запознаване на учениците/студентите с основните аспекти, предмета и предимствата на РНР, както и с информация за тематиката на курса, с текуща информация от областта на учебния процес. С всичко това се цели оптимизиране и повишаване степента на успеваемост на обучаемите.

Системата е проектиран административен модул, който предоставя на потребителите с администраторски права възможност сами да добавят, редактират и изтриват части от системата като по този начин могат да променят структурата ѝ.

Web базираната система предоставя 20 урока по РНР, 5 упражнения по РНР, 1 тест по РНР, форум, в който потребителите могат да дискутират и да задават въпроси по интересни за тях теми.

Задачите, които си поставя настоящата дипломна работа са:

- изграждане на помощен и справочен уеб сайт за обучение по РНР;
- анализ, адекватно и ясно представяне на същността и тематичното съдържание на електронното учебно помагало;

- по-подробно описание на основните теми от лекционния курс, чрез което да се постигне яснота по въпроси, интересувачи както вече запознатите с PHP ученици/студенти, така и потенциални кандидати за обучение в тази област.

- чрез подходящи примери и разнообразни тестове, обучаващите да затвърдят знанията си в изучаваната област.

Предложената от дипломната работа WEB базирана система не претендира за пълнота и изчерпателност. Тя предлага само един метод с тенденцията той да бъде усъвършенстван и доразвит, с което да се постигне възможност за реалната му приложимост в учебния процес.

ГЛАВА 2. ПРОЕКТИРАНЕ НА СИСТЕМАТА

2.1.Организация на системата

WEB версията на дипломната работа може да се раздели основно на три части: информационна, потребителска и администраторска.

Информационната част представлява страници, предоставящи основна информация за структурата, синтаксиса и семантиката на езика, и принципите на програмиране с PHP. Предоставя се форум, в който потребителите изказват мнения и задават въпроси по актуални теми. Информационната част е предназначена за потребители, които само разглеждат сайта – без да се регистрират.

Потребителската част предоставя на всеки потребител както и информацията, съдържаща се в уроците, упражненията, тестовете и форума, така и възможност за добавяне на нови упражнения, и изказване на мнения и задаване на въпроси по темите във форума. Тази част е достъпна за всички потребители осъществили успешно регистриране в сайта.

Последната част е предназначена само за отделна част от потребителите (само за тези, които притежават администраторски права). Тя дава пълна информация, както и възможност за промяна на самия сайт:

- добавяне на нови групи във форума;
- добавяне на нови подгрупи във форума;
- редактиране на вече съществуващи подгрупи;
- изтриване на подгрупи;
- добавяне на нови тестове;
- добавяне на нови въпроси към конкретен тест;
- редактиране на вече съществуващи въпроси;
- изтриване на въпроси;

Чрез използването на този администраторски интерфейс се осъществява лесен достъп, контрол и поддръжка на системата дори и от служители с минимални познания в областта на WEB базираните системи.

2.2. Проектиране на базата от данни

Таблиците са разделени в четири групи:

В първата са таблиците свързани основно с информационната част – уроци и упражнения, това са съответно таблиците lesson и exercises.

Във втората група са таблицата, чрез която се администрира достъпа до системата: users.

В третата група влизат таблиците свързани със създаването на групи, подгрупи и постове във форума. Това са съответно таблиците group_cat, cat и posts.

В последната група влизат таблиците свързани със създаването на тестове и въпроси. Това са таблиците tests и questions.

Структура на таблиците:

lesson – съдържа информацията за уроците:

идентификационен номер на урока (който е уникален), заглавие и съдържание на самия урок.

	Поле	Тип	Колация	Атрибути	Празно	По подразбиране	Допълнително	Действие
	<u>lesson_id</u>	int(11)			не	None	auto_increment	
	lesson_name	varchar(250)	utf8_general_ci		не	None		
	lesson_content	text	utf8_general_ci		не	None		

exercises – е аналогична по структура с lesson:

идентификационен номер (който е уникален), заглавие и съдържание на самото упражнение.

Дипломна работа: „Електронен самоучител по PHP”

	Поле	Тип	Колация	Атрибути	Празно	По подразбиране	Допълнително	Действие
☐	<u>exercises_id</u>	int(11)			не	None	auto_increment	     
☐	exercises_name	varchar(250)	utf8_general_ci		не	None		     
☐	exercises_content	text	utf8_general_ci		не	None		     

users - съдържа основната информация за всеки потребител:

потребителско име, парола (която се съхранява хеширана, за по голяма сигурност), истинско име, e-mail, дата на направената регистрация, вида на регистрацията (потребителска или администраторска) и активността на потребителя.

	Поле	Тип	Колация	Атрибути	Празно	По подразбиране	Допълнително	Действие
☐	<u>users_id</u>	int(11)			не	None	auto_increment	     
☐	login	varchar(50)	utf8_general_ci		не	None		     
☐	pass	varchar(32)	utf8_general_ci		не	None		     
☐	real_name	varchar(100)	utf8_general_ci		не	None		     
☐	mail	varchar(100)	utf8_general_ci		не	None		     
☐	date	int(11)			не	None		     
☐	type	tinyint(1)			не	1		     
☐	activa	tinyint(1)			не	1		     

group_cat – съдържа основната информация за групите от форума:

идентификационен номер на групата, име, нейната активност, дата на създаване и кратко описание.

	Поле	Тип	Колация	Атрибути	Празно	По подразбиране	Допълнително	Действие
☐	<u>group_cat_id</u>	int(11)			не	None	auto_increment	     
☐	name	varchar(100)	utf8_general_ci		не	None		     
☐	active	tinyint(1)			не	1		     
☐	date_added	int(11)			не	None		     
☐	desc	varchar(250)	utf8_general_ci		не	None		     

cat – съдържа основната информация за подгрупите от форума:

Дипломна работа: „Електронен самоучител по PHP”

собствен идентификационен номер, идентификационен номер на групата, към която принадлежи, име, кратко описание, активността ѝ и дата на създаване.

	Поле	Тип	Колация	Атрибути	Празно	По подразбиране	Допълнително	Действие
☐	<u>cat_id</u>	int(11)			не	None	auto_increment	      
☐	group_cat_id	int(11)			не	None		      
☐	name	varchar(100)	utf8_general_ci		не	None		      
☐	desc	varchar(250)	utf8_general_ci		не	None		      
☐	active	tinyint(1)			не	1		      
☐	date_added	int(11)			не	None		      

posts – съдържа информация за всеки пост:

идентификационен номер, идентификационен номер на подгрупата, към която принадлежи, от кого и кога е добавен, заглавие, съдържание на самия пост, колко пъти е видян и от кого, и кога е редактиран поста.

	Поле	Тип	Колация	Атрибути	Празно	По подразбиране	Допълнително	Действие
☐	<u>post_id</u>	int(11)			не	None	auto_increment	      
☐	cat_id	int(11)			не	None		      
☐	added_by	int(11)			не	None		      
☐	date_added	int(11)			не	None		      
☐	title	varchar(100)	utf8_general_ci		не	None		      
☐	content	text	utf8_general_ci		не	None		      
☐	view_count	int(11)			не	None		      
☐	edited_by	int(11)			не	None		      
☐	edited_when	time			не	None		      

tests – съдържа основна информация за тестовете:

идентификационен номер, име, кратко описание и автор на теста.

	Поле	Тип	Колация	Атрибути	Празно	По подразбиране	Допълнително	Действие
☐	<u>id</u>	int(12)			не	None	auto_increment	      
☐	ime	varchar(50)	utf8_unicode_ci		не			      
☐	opisanie	mediumtext	utf8_unicode_ci		не	None		      
☐	avtor	varchar(25)	utf8_unicode_ci		не			      

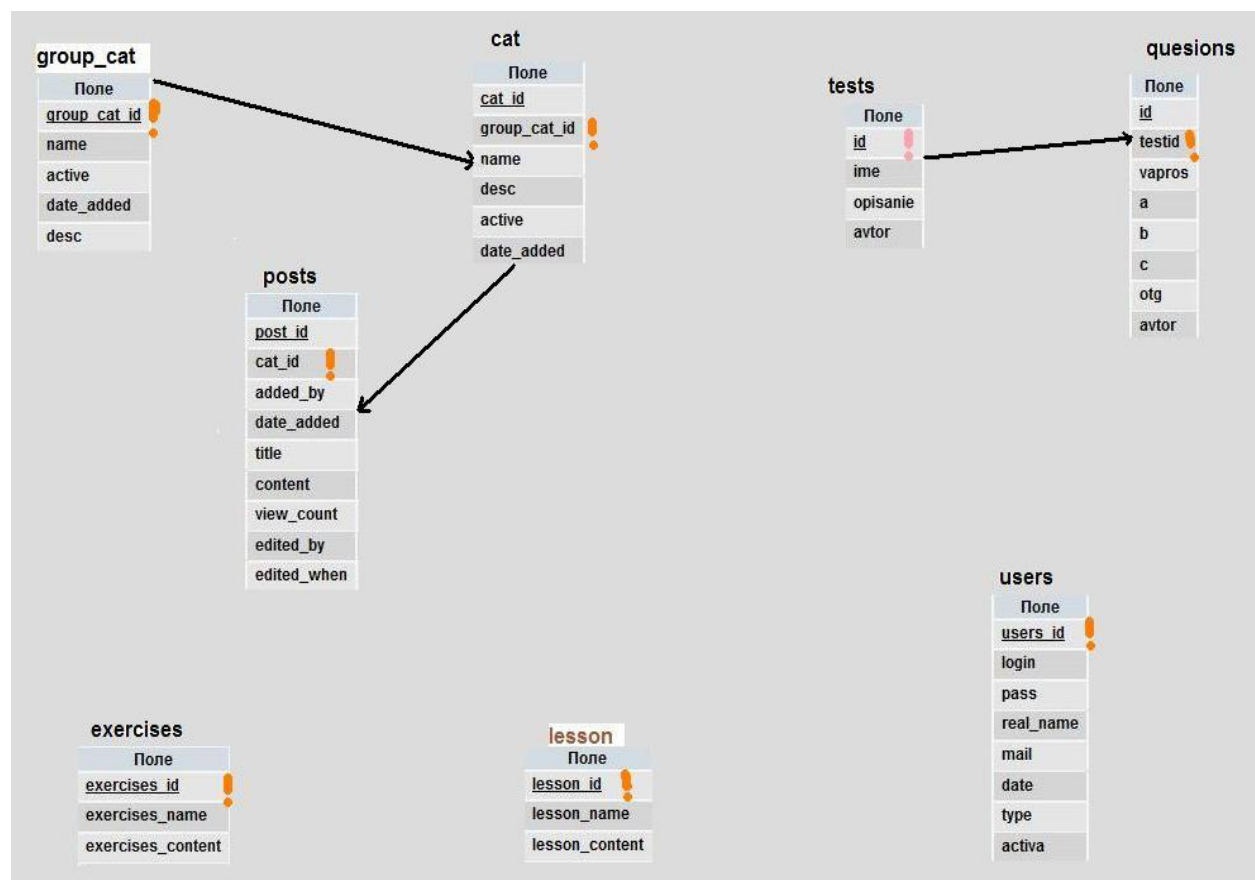
Дипломна работа: „Електронен самоучител по PHP”

questions – съдържа информация за въпросите от тестовете:

идентификационен номер, идентификационен номер на теста, към който принадлежи, въпроса, трите възможни варианти за отговор, самия отговор и автор на въпроса.

	Поле	Тип	Колация	Атрибути	Празно	По подразбиране	Допълнително	Действие
☐	id	int(12)			не	None	auto_increment	      
☐	testid	int(12)			не	0		      
☐	vapros	varchar(130)	utf8_unicode_ci		не			      
☐	a	varchar(40)	utf8_unicode_ci		не			      
☐	b	varchar(40)	utf8_unicode_ci		не			      
☐	c	varchar(40)	utf8_unicode_ci		не			      
☐	otg	varchar(40)	utf8_unicode_ci		не			      
☐	avtor	varchar(25)	utf8_unicode_ci		не			      

Структура на връзките между таблиците в базата данни:



ГЛАВА 3. ПРОГРАМНА РЕАЛИЗАЦИЯ

3.1. Използвани технологии.

3.1.1. HTML (HyperText Markup Language)

HTML представлява език за форматиране на хипертекст, даващ възможност за визуализиране на текст, график в браузърите или други приложения.

Това е описателен език за форматиране, чрез който се създават документи, независещи от хардуерната или софтуерната платформа. Основните възможности на HTML са :

- Оформление на документи
- Включване на връзки към други Интернет страници
- Включване на графики, звук
- Създаване на таблици и други
- Създаване на списъци

Езикът за форматиране на WEB страници HTML е универсалния език във WEB. Той се използва за създаване на статични страници (в последните версии на езика, той е допълнен с т. нар. DHTML - Dynamic HyperText Language, който добавя известна интерактивност на страниците) и представлява обикновени ASCII файлове т. е. това са обикновени текстови файлове, които може свободно да се променят с помощта на Notepad. Тези файлове се състоят от текст, който бива

визуализиран и команди (тагове, tags), които служат за указване на това как да бъде визуализиран текста.

3.1.2.CSS (Cascading Style Sheets)

CSS е език за описание на стиловете - използва се основно за описване на представянето на документ, написан на език за маркиране. Най – често се използва заедно със HTML, но може също така да се приложи и върху

XHTML, XML и производните му. Официално спецификацията на CSS се поддържа от W3C.

Създаден първоначално като средство за разделяне на съдържанието от представянето му, днес той се използва основно за визуално оформление на HTML страници.

Каскадните стилове позволяват в много по-добра степен контрол върху елементите на страницата, като добавят и нови възможности за форматиране на страницата.

Положителния ефект от използването на каскадни стилове е този, че ако се наложи да се промени дадено форматиране е достатъчно да се промени само на едно място, а именно файла съдържащ каскадния стил.

3.1.3. SQL - Език за структурирани запитвания (Structured Query Language)

SQL е непроцедурен език за създаване на заявки към база от данни.

Непроцедурен означава „какво, а не как”. Например SQL описва какви данни трябва да бъдат извлечени, изтрети или вмъкнати, а не как трябва да бъде извършена тази операция. SQL е разработен, за да обслужва релационните бази от данни. Всъщност може да се програмира на SQL само за релационни бази от данни, които поддържат SQL, като MSAccess, Oracle, Sybase и Informix. Въпреки, че реализациите на всеки потребител се различават един от друг, това не води до съществени изменения на SQL кода.

Една от най – мощните способности на SQL е възможността да събира и обработва данни от няколко таблици едновременно.

Едно от големите преимущества на SQL е, че е напълно платформено и продуктово независим език. Той е от така наречените езици от високо ниво или четвърто поколение програмни езици, при които може да се свърши много работа с малко код. SQL притежава много мощна система от команди, с които могат да се извършат всички необходими операции с бази от данни – от създаването на база от данни до манипулирането на данните и създаване на вложени процедури.

SQL е стандартният език, използван за манипулиране и извличане на данни от релационните бази данни. SQL позволява на програмиста или администратора на бази данни да извърши следните неща:

- Модифициране на структурата на базите данни
- Промяна на системните настройки за сигурността
- Добавяна на разрешения за достъп на даден потребител до бази данни или таблици
- Разпитване на базата данни за информация
- Обновяване съдържанието на базата данни

Синтаксиса на SQL е много гъвкав, въпреки че има правила, които трябва да се спазват, както е при всеки програмен език.

Командата *Select* се използва много по-често от всяка друга SQL команда. Тя е незаменима при всяка комуникация с компютъра, отнасяща се за извличане на данни.

3.1.4.MySQL

MySQL е open source релационна, управляваща бази от данни система (RDBMS), която използва SQL, най – популярния език за въвеждане, достъп и извършване на други процеси с данни в база данни. Това е open source, всеки може да изтегли MySQL и да добави към нея, в зависимост от общите публични правила. MySQL е известен основно заради скоростта си, сигурността и гъвкавостта си.

Системата за релационни бази данни MySQL, осигурява приложен програмен интерфейс (API) за C, C++, Java, Perl, PHP, Python и Tcl, допуска функциите в SELECT и WHERE частите на заявките.

Понастоящем MySQL работи върху Linux, UNIX и Windows платформи. Голям е интереса към MySQL като алтернатива за подходящи

системи бази данни от Oracle, IBM и Informix.

Невъзможно е да си представим модерен и високофункционален Уеб сайт, в основата на който не стои база данни. Множество от функциите, които правят страниците привлекателни не биха могли да съществуват, услуги като онлайн банкиране, пазаруване и дори просто организиране на информацията, са немислими без някакъв вид система за бази данни. Няма да навлизаме в технологични и терминологични подробности, а ще покажем как да използвате на практика една от най-популярните и разпространени в Интернет системи за управление на бази данни - MySQL.

Основната версия на системата е безплатна за използване, без значение с комерсиални или идеални цели, като това е една от основните причини за широката ѝ популярност. Това не означава, че възможностите, които предлага са недостатъчни. Системата може да изпълнява заявките на неограничен брой потребители, издържа до 50 милиона записа, предлага лесна за използване система за управление на правата на различните потребители, отлична скорост на обработка на заявките.

3.1.5. PHP

PHP е скриптов език, който стана известен като компонент на много средно големи и големи уеб приложения за бази от данни. Създаден като личен проект, той бива преработен от група разработчици и бързо намира приложението си в Интернет. За да улесни грижите си за собствения си личен сайт, през 1994 година Размус Лердорф създава скриптов език, наричайки го Personal Home Page Tools. От това наименование по-късно излиза и съкращението PHP.

Нововъведението бързо събужда интереса на множество разработчици и скоро след това първата версия на езикът е вече факт. Съкращението PHP вече

се разбира като PHP: Hypertext Preprocessor, но основната идея - простотата на използване, е запазена. PHP и до днес остава силен скриптов език с много възможности за web програмиране. И може би най-лесния за изучаване и прилагане.

Факт е, че PHP позволява да постигнете определен резултат само с няколко реда код, докато ако използвате други програмни езици за целта може да се наложи да изпишете десетки редове. Това е основната причина, поради която езикът получи толкова бърза и голяма популярност сред web разработчиците. Използването на PHP спестява време, а и позволява създаването на интерактивни сайтове дори от начинаещи. Малко по - малко почти всички хостващи компании започват да предлагат поддръжка за езика. PHP е избран поради няколко причини:

- PHP е с отворен код и е безплатен.
- В статичните HTML - файлове могат да се вграждат един или повече PHP – скриптове и това улеснява интегрирането на клиентския слой.
- Съществуват повече от 15 библиотеки за бърз достъп до слоя бази данни.
- Бързи изпълнения на скриптовете. Изпълнението става бързо и всички компоненти работят в главната памет на PHP (за разлика от скриптови платформи, при които компонентите се намират в отделни модули). Доказано е, че за задачи от средна сложност PHP е по – бърз от останалите популярни скриптов езици.
- Гъвкавост на платформите и опрационните системи. Apache работи на много различни платформи и под избрани операционни системи;

Дипломна работа: „Електронен самоучител по PHP”

PHP работи както на всички тях, така и на много други, когато е интегриран с останалите уеб сървъри.

- PHP е подходящ за сложно системно разработване. Той е пълен език за програмиране, притежаващ повече от 50 функционални библиотеки.

Предимства на езика:

- PHP работи на множество операционни системи (Unix, Linux, Windows, BSD, Mac OS X) и множество уеб сървъри – Apache , lighttpd, IIS
- PHP е лесен за разработване
- PHP е безплатен и се разпространява под лиценза на BSD
- PHP може да бъде лесно модифициран и адаптиран към нуждите на прилагащата го организация
- PHP е създаден и пригоден за разработката на уеб приложения
- PHP не изисква особени умения от разработчици работили на структурни езици - езикът е с прост и интуитивен синтаксис за такива разработчици
- PHP е широко разпространен поради простотата си. Има много програмисти, което води до по - евтино платен персонал във фирмите, по ниска цена на приложенията за клиентите и още по-голяма използваемост. Поддръжката за PHP разработчици е гарантирана от множеството форуми и приложения на общността.
- По аналогия с Perl към стандартните класове на PHP могат да бъдат писани и много допълнителни модули
- PHP поддържа следните системи за управление на бази от данни:

IBM DB2 - formix - Ingres, Microsoft SQL Server (MS SQL), mSQL, MySQL, Oracle, PostgreSQL, Sybase

- PHP поддържа и ODBC

Критики:

- В по-старите версии на езика по подразбиране съществува функционалност, чрез която крайните потребители могат да подават направо променливи към PHP скрипта (`register_globals`). Настройката по подразбиране е премахната от версия 4.2.0, но съществуват все още много стари приложения, които предполагат употребата на тази функция.
- PHP не поддържа напълно Unicode. Пълната поддръжка се предвижда в PHP 6.
- PHP не поддържа именни пространства - т. е. всички функции се намират в едно именно пространство.
- Съществуват много функции на библиотеки с идентична функционалност.
- Няма изградена конвенция за именуване и подреждане на аргументите на вградените функции в PHP, което прави необходимо постоянно консултиране с документация при работа с PHP.
- PHP няма вградена поддръжка на нишки (`threads`), което прави обработката на големи масиви от данни много трудна: страницата не се зарежда преди целият масив от данни да е преработен и ако потребителят прекъсне зареждането (и уеб сървърът разрешава прекъсване на заявката), прекъсва и обработката на данните.
- Обръщенията към неинициализирани променливи и елементи на масив са разрешени в PHP, което прави възможно допускането на трудни за откриване правописни грешки.
- PHP използва т. нар. "magic quotes", които са създадени с цел да предпазват начинаещите програмисти, но объркват по-напредналите.

- Понякога се среща неконсистентност между поведението на PHP в различните версии, което предизвиква необходимостта от пренаписване на скриптовете, писани за стари версии на езика.

Управление на сесиите чрез php:

Управлението на сесиите в PHP е механизъм, който позволява да се направи връзка между запитванията на даден клиент от дадено време. Както знаеме HTTP - протокола не прави връзка между различните запитвания, и след всяко ново запитване този процес приключва, докато не бъде направено ново запитване, което да стартира нов процес. Така за даден клиент се губи релацията между неговите запитвания. Затова се появява нуждата от управление на сесиите, което да позволи съответната релация между запитванията на даден клиент. При това се създават съответните Session - IDs автоматично, и данните се запазват и съхраняват от управлението на сесиите. PHP използва в случая на запазване на данните от сесията файловата система на съответния сървър. При запазване на данните променливите се сериализират автоматично.

При стартиране на дадена сесия се проверява първо, дали съществува правилен Session - ID. Ако няма такъв, то тогава php прави съответния id. Ако има вече сесия, то тогава променливите на сесията могат да се използват като глобални променливи в дадения скрипт.

Свързване към база от данни MySQL посредством PHP

Свързването на MySQL и PHP е петстепенен процес:

1. Свързване към DBMS и използване на базата от данни. Връзката към MySQL се отваря посредством `mysql_connect()`. Тази PHP команда има три параметъра: името на сървъра за бази от данни, който ще се използва, потребителското име и парола. След като се осъществи връзката към сървъра с командата `mysql_select_db()` се избира базата от данни, която ще се използва.

Функцията `mysql_connect()` връща манипулатор на връзката. Манипулаторът е стойност, която може да се използва за получаване на достъп до информацията, асоциирана с връзката.

2. Стартиране на заявката. Това става посредством `mysql_query()`. Тази функция има два параметъра: самата заявка на SQL и връзката към DBMS, която трябва да се използва. Параметъра представляващ връзката е стойността, върната при създаването на връзката в степен 1. Функцията `mysql_query()` връща стойност, която може да обработи изходния резултат от степен 3.

3. Обработване на резултатите. Функцията `mysql_fetch_row()` обработва по един ред от резултатите, вземайки като параметър само резултатния манипулатор от стъпка 2. Всеки ред се записва в масив `$row`, а стойностите на атрибутите в масива се извличат в стъпка 4.

Обикновено редовете се извличат чрез цикъл `while`. Когато няма повече налична информация `mysql_fetch_row()` връща `false`.

4. Обработване на стойностите на атрибутите. За всеки обработен ред се използва цикъл `for`, който извежда всеки атрибут от текущия ред посредством инструкция `echo`. Функцията `mysql_num_fields()` се използва за намиране броя атрибути в реда, т. е. на броя елементи в масива.

5. Затваряне на връзката към DBMS посредством `mysql_close()`. Тази функция няма параметри.

3.1.6. JavaScript

Javascript е най-широко разпространеният език за програмиране в интернет, след HTML. Въпреки че го нарекох "език за програмиране", с негова помощ не се пишат програми, а скриптове които се вмъкват в HTML документа. В този смисъл JavaScript е език за писане на скриптове, докато JAVA е език за програмиране. Освен съвпадението в част от името, двата езика нямат кой знае какви прилики, дори са разработени от различни корпорации (JAVA е дело на SUN, а JavaScript е разработка на Netscape). JAVA е мощен език за програмиране не само на интернет приложения, но и на самостоятелни програми за различни платформи. Интернет приложенията на JAVA се наричат аплети. Те са файлове с разширение `.class` и се вмъкват в HTML документа между таговете `<APPLET>` и `</APPLET>`. Тук няма да се спираме подробно на JAVA аpletите.

Нека разгледаме възможностите на JavaScript, какво можете и какво не можете да правите с него:

- Ефекти с изображения. Rollover ефекти, слайд шоу, и много други.
- Управление на прозорци и рамки. Отваряне и затваряне на прозорци, задаване на размера на прозорец, управление на един прозорец от друг и т. н.

- Разпознаване на типа на браузъра, операционната система, разделителната способност на екрана и дълбочината на цветовете.
- Много други работи

Какво не можете да правите с помощта на JavaScript:

- Не можете да записвате информация на сървъра (не можете да организирате форуми, да обработвате бази данни)

JavaScript кода се вмъква в HTML документа между двойката елементи `<SCRIPT>` и `</SCRIPT>`. Когато срещне тагът `<SCRIPT>`, браузъра разбира че трябва да спре интерпретирането на HTML кода и да започне да обработва скрипта, намиращ се между `<SCRIPT>` и `</SCRIPT>`. Този скрипт не е задължително да бъде написан на JavaScript. Има и други езици за писане на скриптове, например VBScript. Затова когато пишете отварящия таг за скрипт, трябва да укажете на браузъра на какъв език ще бъде скрипта. Ако този език е JavaScript трябва да напишете `<SCRIPT LANGUAGE="JavaScript">`

3.1.7 APACHE

Apache HTTP Server е уеб сървър с отворен код за Unix, Windows, Novell NetWare и други платформи. Apache има огромен принос за първоначалното разрастване на WWW и чрез него работят над 70% от сайтовете (включително Wikipedia). Счита се от много специалисти за платформа, според която се разработват и оценяват другите уеб сървъри.

Сървърът има възможности за промяна на съобщенията за грешки, удостоверяване на потребителите, договаряне на съдържанието (изключително полезно при многоезични сайтове), проху възможности, както и поддръжка на CGI и SSI. Има множество модули за Apache, които позволяват работа на

разнообразни скриптове и осигуряване на динамично съдържание, криптиране, ограничаване и други.

Apache се разработва от отворено общество от разработчици - Apache Software Foundation. Обществото поддържа и разработва множество други проекти най-важните от които са Apache Ant, Apache SpamAssassin, Apache Tomcat и огромен брой проекти основани на XML езика, но Апачи е най-известния и широко разпространен продукт.

3.2. Описание на реализираните на PHP страници съставлящи WEB сайта:

Index.php

Първата страница, на която попада потребителя след зареждане на уеб адреса в браузъра е страница „My PHP - начало” (index.php).

В горната си част съдържа името, рекламната картинка на PHP и навигационна лента за връзка към всички останали страници. Това се реализира с помощта на HTML тагове необходими за визуализацията на сайта на клиентския браузър. В конкретният случай, обаче тези HTML тагове не са използвани директно във файла, а се извикват като функции на друг файл - **functions.php**.

functions.php

Този php файл се извиква във всяка една страница. Както подсказва и самото му име, той съдържа функции, необходими за реализирането на останалите страници на приложението. Състои се от четири функции, две от които се извикват във всеки файл, в зависимост от правата, които има дадения потребител. Това са функциите my_header и my_adminheader.

.....

```
function my_header($title) {
    require_once('header.php');
    require_once('left.php');
?>

<html>
<head>
    <title><?php echo $title ?></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" href="css/css.css" type="text/css"/>
    <script type="text/javascript" src="js/prototype.js"></script>
<script type="text/javascript" src="js/common.js"></script>
</head>
<body><div id="list">

    <?php
    if ($_SESSION['is_logged'] === true) {
        echo 'Здравей, <b>' . $_SESSION['user_info']['login'] . '</b>';
        if ($_SESSION['user_info']['type'] == 3) {
            }
        }

    ?>
</div>

<div id="content">

<?php
```

```
}
```

.....

Функцията `my_header` съдържа отварящите HTML тагове, инициализира името на потребителя, ако той е регистриран и извиква файла `left.php` - менюто на приложението, което е достъпно за нерегистрираните и за регистрираните потребители, които нямат администраторски права.

.....

```
function my_adminheader($title) {
```

```
    require_once('header.php');
```

```
    require_once('left.php');
```

```
?>
```

```
<html>
```

```
<head>
```

```
    <title><?php echo $title ?></title>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
    <link rel="stylesheet" href="css/css.css" type="text/css"/>
```

```
    <script type="text/javascript" src="js/prototype.js"></script>
```

```
<script type="text/javascript" src="js/common.js"></script>
```

```
</head>
```

```
<body><div id="list">
```

```
    <?php if ($_SESSION['is_logged'] === true) {
```

```
        echo 'Здравей, <b>' . $_SESSION['user_info']['login'] . '</b>';
```

```
    }
```

```
        if ($_SESSION['is_logged'] !== true &&
$_SESSION['user_info'][type] != 3) {
```

```
            header('Location: index.php');
```

```
        exit;
    }
    ?>

        <ul id="navmenu">
        <li><a href="groups.php">Групи форуми</a></li>
        <li><a href="sub_groups.php">Под групи форуми</a></li>
        <li><a href="addtest.php">Нов тест</a></li>
        <li><a href="addquest.php">Нови въпроси</a></li>
        <li><a href="questions.php">Редактирай въпроси</a></li>
        </ul> </div>

<?php
    }
```

Функцията `my_adminheader` е еквивалентна на `my_header`, в допълнение обаче: прави се проверка, дали потребителя има администраторски права и ако е така, се инициализира администраторското меню.

Следват функциите `db_init` и `run_q`. Първата функция служи за връзка с базата данни, а втората се извиква преди всяка SQL заявка, като нейната цел е да реши част от проблемите с енкодингите.

register.php

Състои се от HTML форма за регистрация, която има пет полета за попълване на лична информация от потребителите, като за всяко поле по отделно се прави проверка за непозволени символи, дължина. Второто и третото поле са отделени за попълване на парола (която се запазва в базата данни в хеширан вид), като в

този случай проверката е дали съвпадат двете полета. Четвъртото поле – E - mail, трябва да отговаря на по строги изисквания – проверяваме дали в текста се съдържа @ и след това точка. Тази проверка е реализирана чрез регулярен израз:

```
if (!strstr("$mail", "@") || !strstr("$mail", ".")) {  
    $error_array['$mail'] = 'Невалиден адрес';  
}
```

Ако са минали тези проверки успешно следва проверка в базата данни – дали има други регистрации със същото потребителско име или адрес. Ако и тази проверка премине успешно, тогава информацията получена от формата се попълва в базата данни и регистрацията е успешна, в този случай потребителя се пренасочва към началната страница.

login.php

Състои се от форма от две полета за попълване – потребителско име и парола. След попълването ѝ се прави справка в базата данни, дали съществува такъв потребител с такава парола и ако има съвпадение, потребителя отново се пренасочва към началната страница, като се стартира сесия чрез която се предава потребителското име на потребителя.

ind.php

Съдържа списък с уроците, като всеки урок, в зависимост от идентификационния номер, представлява хипер връзка към конкретния урок.

```
$rs = run_q('SELECT * FROM lesson ');  
while ($row = mysql_fetch_assoc($rs)) {  
    echo '<div class="content1"><br/>';
```

```
echo '<div class="link"><a href="lesson.php?id=' . $row['lesson_id'] . "'>' .  
$row['lesson_name'] . '</a></div>';  
}
```

lesson.php

Осъществява връзка с базата данни и извлича от нея заглавието на урока и неговото съдържание.

exercises.php и excontent.php

Аналогични са съответно на **ind.php** и **lesson.php**, като в този случай се извлича информация за упражненията. В случай, че потребителя, който разглежда упражненията, е регистриран, тогава под списъка с уроците в **exercises.php** се реализира HTML форма. Тя позволява на потребителя да добави свое собствено упражнение.

tests.php

Извлича наличните тестове от базата данни и прави проверка на дадените отговори на теста. Като след решаване на целия тест извършва цялостна справка за количеството верни и грешни отговори и съобщава резултата на потребителя, като при грешните отговори се съобщава и номера на сгрешения въпрос.

.....

```
db_init();  
$id = $_GET['id'];  
if ($id) {  
$sql = "SELECT * FROM questions WHERE testid = '$id' ORDER BY id ASC";  
$res = mysql_query($sql) or die(mysql_error());
```

```
if (!$_POST['submit']) {
echo "<form name='test' method='POST'>";

$vapros = 1;
while ($row = mysql_fetch_array($res)) {
echo "$vapros. $row[vapros]";
echo "<br><input name='$row[id]' type='radio' value='a'>".$row[a];
echo "<br><input name='$row[id]' type='radio' value='b'>".$row[b];
echo "<br><input name='$row[id]' type='radio' value='c'>".$row[c]."<br>";
echo '<br/>';
$vapros++;
}

echo "<br><input type='button' onclick='javascript: history.back(-1);' name='back'
value='Назад'>

<input type='submit' name='submit' value='Провери'> <input type='reset'
name='reset' value='Наново'><br>";
echo "</form>";
}

else {
$veren = 0;
$gre6en = 0;
$i = 0;
while($row = mysql_fetch_array($res)) {
$id = $row[id];
$otg = $_POST[$id];
$verniq = $row[otg];
```



```
if ($row[otg] == $otg) {
    $i++;
    $veren++;
    echo "<span class='true'>Върпос $i: Вярно!</span><br>";
} else {
    $i++;
    echo "<span class='false'>Върпос $i: Грешно! Верния отговор е: $row[otg] - $row[$verniq]</span><br>";
    $greben++;
} }
echo "Верни отговори: ".$veren;
echo "<br>Грешни отговори: ".$greben;
echo "<br>";
$all = $veren + $greben;
$koef = $veren/$all;
$koef = $koef*100;
$koef = round($koef);
$id = $_GET['id'];
$zaqvka = "SELECT * FROM tests WHERE id = '$id'";
$result = mysql_query($zaqvka) or die (mysql_error());
$row1 = mysql_fetch_array($result);
$test = $row1[ime];
echo "Вашия резултат е: $koef%<br>";
}
}
```

```
else {  
$sql ="SELECT * FROM tests";  
$res = mysql_query($sql) or die(mysql_error());  
while($row = mysql_fetch_array($res)) {  
echo "<a href='tests.php?id=$row[id]'>$row[ime]</a><br>  
$row[opisanie]<br><br>";  
}    }  
.....
```

findex.php

Извежда списък на групите и техните активни подгрупи (теми) във форума, като всяка подгрупа е хипер връзка към съответната тема.

.....

```
<a href="topic.php?id=' . $row['cat_id'] . "'>' . $row['name'] . '</a><p>' . $row['desc'] .  
'</p>';
```

.....

topic.php

Файлът е аналогичен на lesson.php и excontent.php. Извлича всички постове, писани по конкретната тема, като освен това извлича и името на потребителя и датата, на която е писан поста. Извършва се проверка и, ако потребителя е регистриран, се допълва навигационното меню, като на потребителя се позволява да запише свой собствен пост.

post.php

Реализира HTML форма, която позволява на потребителя да въведе заглавие и съдържание на своя пост, като след попълване на формата се извършва проверка за дължината на текста и, ако проверките преминат успешно, поста се записва в базата данни.

groups.php

Този файл е достъпен само за потребители с администраторски права. Той извлича наличните групи форуми от базата данни и позволява на администратора да редактира съществуващи теми и да създава нови.

```
.....
db_init();
if ($_POST['ng'] == 1) {
    $name = addslashes(trim($_POST['group_name']));
    $desc = addslashes(trim($_POST['desc']));
    if (strlen($name) > 4) {
        $id = (int) $_POST['edit_id'];
        $rs = run_q('SELECT * FROM group_cat WHERE name="" . $name . "" AND
group_cat_id !=' . $id);
        if (!mysql_numrows($rs) > 0) {
            if ($id > 0) {
                run_q('UPDATE group_cat SET name="" . $name . "", `desc`="" . $desc . ""
WHERE group_cat_id=' . $id);
                echo '<h1>Успешно обновяване!</h1>';
            } else {
```

```
run_q('INSERT INTO group_cat(name, date_added, `desc`)
VALUES("'" . $name . "', ' . time() . ', '" . $desc . "'");

echo '<h1>Успешен запис!</h1>';
}
} else {
    echo '<h1>Името съществува!</h1>';
}
}
}

echo '<div class="content">';

$rs = run_q('SELECT * FROM group_cat');

echo '<br/><table
border="1"><tr><td>Име</td><td>Описание</td><td>Редактирай</td></tr>';

while ($row = mysql_fetch_assoc($rs)) {
    echo '<tr><td>' . $row['name'] . '</td><td>' . $row['desc'] . '</td>
    <td><a href="groups.php?mode=edit&id=' . $row['group_cat_id'] .
    "'>Редактирай</a></td></tr>';
}

if ($_GET['mode'] == "edit" && $_GET['id'] > 0) {
    $id = (int) $_GET['id'];

    $rs = run_q('SELECT * FROM group_cat WHERE group_cat_id=' . $id . '');

    $ed_info = mysql_fetch_assoc($rs);
}

echo '</table><br/><br/><br/>';
```

```
echo '<form method="POST" action="groups.php">
```

```
    Име на група:<input type="text" name="group_name" value="" . $ed_info['name'] . ""  
><br>
```

```
    Описание:<textarea name="desc" rows="5" cols="50" >' . $ed_info['desc'] .  
'</textarea><br>
```

```
    <input type="submit" value="Запиши">
```

```
    <input type="hidden" name="ng" value="1">;
```

```
if ($_GET['mode'] == "edit") {
```

```
    echo '<input type="hidden" name="edit_id" value="" . $_GET['id'] . "">;
```

```
}
```

```
.....
```

sub_groups.php

Аналогичен е на **groups.php**, с разликата че тук администраторът трябва да избере към коя група принадлежи подгрупата която ще създава.

```
$rs = run_q('SELECT * FROM group_cat');
```

```
echo '<form method="POST" action="sub_groups.php">
```

```
    Група:<select name="group">;
```

```
while ($row = mysql_fetch_assoc($rs)) {
```

```
    if ($row['group_cat_id'] == $ed_info['group_cat_id']) {
```

```
        echo '<option value="" . $row['group_cat_id'] . "" selected="selected">' .
```

```
$row['name'] . '</option>;
```

```
    } else {
```

```
        echo '<option value="" . $row['group_cat_id'] . "">' . $row['name'] . '</option>;
```

```
    }
```

```
}
```

addtest.php и addquest.php

Реализират HTML форми. Първата позволява създаването на нов тест, като формата се състои от две части – заглавие и кратко описание на теста. Втората е аналогична на sub_groups.php. Позволява на администратора да избере към кой тест да добавя въпрос. Прави проверка дали е записан кой е верния отговор и ако не е извежда съответното съобщение.

.....

```
$sql = "SELECT id FROM questions ORDER BY id DESC";
```

```
$res = mysql_query($sql);
```

```
$row = mysql_fetch_array($res);
```

```
$id = $row[id] + 1;
```

```
$testid = $_POST['testid'];
```

```
$otg = $_POST['otg'];
```

```
if (!$otg) {
```

```
    echo "Посочете верния отговор";
```

```
} else {
```

```
$vapros = $_POST['vapros'];
```

```
$a = $_POST['a'];
```

```
$a = trim(strip_tags(addslashes($a)));
```

```
$b = $_POST['b'];
```

```
$b = trim(strip_tags(addslashes($b)));
```

```
$c = $_POST['c'];
```

```
$c = trim(strip_tags(addslashes($c)));
```

```
$otg = $otg;
```

```
$query = "INSERT INTO questions (testid, vapros, a, b, c, otg)
```

```
VALUES ('$testid', '$vapros', '$a', '$b', '$c', '$otg')";
```

```
$res = mysql_query($query) or die (mysql_error());  
if ($res) { echo 'Въпроса е добавен успешно.<br><a href="javascript:history.go(-  
1)">Назад</a>'; }  
}  
}
```

.....

questions.php

Извлича всички въпроси от базата данни и позволява тяхното редактиране или изтриване. При избор за редактиране, се реализира формата от addquest.php, попълнена с конкретния въпрос. Реализира се по следния начин:

```
.....  
$edit = $_GET['edit'];  
$sql1 = "SELECT * FROM questions";  
$ras1 = mysql_query($sql1) or die (mysql_error());  
$row1 = mysql_fetch_array($ras1);  
$testid = $row1[testid];  
$vapros = $_GET['edit'];  
$sql = "SELECT * FROM questions WHERE id = '$vapros'";  
$res = mysql_query($sql);  
$row = mysql_fetch_array($res);  
$sql2 = "SELECT * FROM tests WHERE id = '$testid'";  
$res2 = mysql_query($sql2);  
$row2 = mysql_fetch_array($res2);  
if ($_POST['redaktirai']) {  
$otg = $_POST['otg'];
```

```
$vapros = $_POST['vapros'];
$a = $_POST['a'];
$b = $_POST['b'];
$c = $_POST['c'];
if ($otg == NULL) {
    $zaqvka = "UPDATE questions SET vapros = '$vapros', a = '$a', b = '$b', c='$c'
    WHERE id = '$edit'";
    $res = mysql_query($zaqvka) or die (mysql_error());
    echo "Въпроса е обновен успешно";
} else {
    $zaqvka = "UPDATE questions SET vapros = '$vapros', a = '$a', b = '$b', c='$c', otg =
    '$otg' WHERE id = '$edit'";
    $res = mysql_query($zaqvka) or die (mysql_error());
    echo "Въпроса е обновен успешно";
}
} else {
    $otg = $row[otg];
    echo "<form name='editform' method='post' action='>
    Тест: $row2[ime] <br>
    Въпрос: <input type='text' name='vapros' value='$row[vapros]'><br>
    Отговор А: <input type='text' name='a' value='$row[a]'><input type='radio' name='otg'
    value='a'><br>
    Отговор Б: <input type='text' name='b' value='$row[b]'><input type='radio' name='otg'
    value='b'><br>
    Отговор В: <input type='text' name='c' value='$row[c]'><input type='radio' name='otg'
    value='c'><br>
```



```
<input type='submit' name='redaktirai' value='Редактирай'><br>
```

Сега верния отговор е **\$otg - \$row[\$otg]**. Ако искате да го смените изберете друг ако ли не не избирайте нищо.";

```
}
```

.....

При решение за изтриване на даден въпрос, се дава възможност за избор дали наистина да бъде изтрит въпроса или не.

.....

```
if ($_POST['submit']) {
```

```
$delete = $_GET['delete'];
```

```
$sql = "DELETE FROM questions WHERE id='$delete'";
```

```
$result = mysql_query($sql) or die (mysql_error());
```

```
echo "Въпроса е изтрит успешно.<br>";
```

```
echo '<input type="button" onclick="javascript: history.go(-2);" name="back" value="Назад">';
```

```
} else {
```

```
echo "<center><div class='confirm'>Сигурен ли сте че искате да изтриете въпроса?<br>
```

```
<form name='delete' method='POST'>
```

```
<input type='submit' name='submit' value='Да'>
```

```
<input type='button' onclick='javascript: history.back(-1);' name='back' value='Не'>
```

```
</form></div></center>
```

```
";      }
```

.....

Във всеки един файл, с изключение на functions.php се стартира сесия, чрез която се запазва информацията за конкретния потребител (ако той е регистриран).

logout.php

Стартира сесията и след това я изключва, и препраща потребителя към началната страница.

```
<?php  
session_start();  
ob_start();  
session_destroy();  
header('Location: index.php');  
exit;  
?>
```

ГЛАВА 4. РЪКОВОДСТВО НА ПОТРЕБИТЕЛЯ

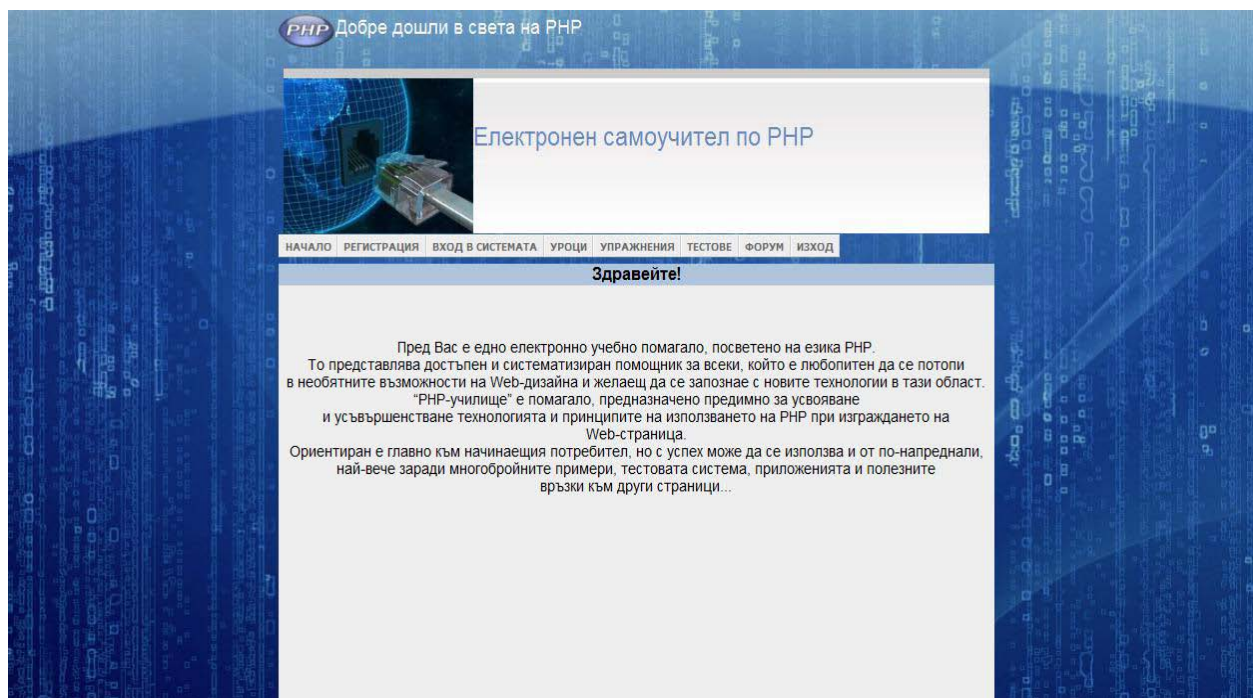
4.1. Системни изисквания

Минималните системни изисквания за изпробването на текущия сайт на локалния компютър са:

- Операционна система Windows 98, Windows ME, Windows XP или Windows Vista, Linux
- Mozilla Firefox

4.2. Описание на сайта

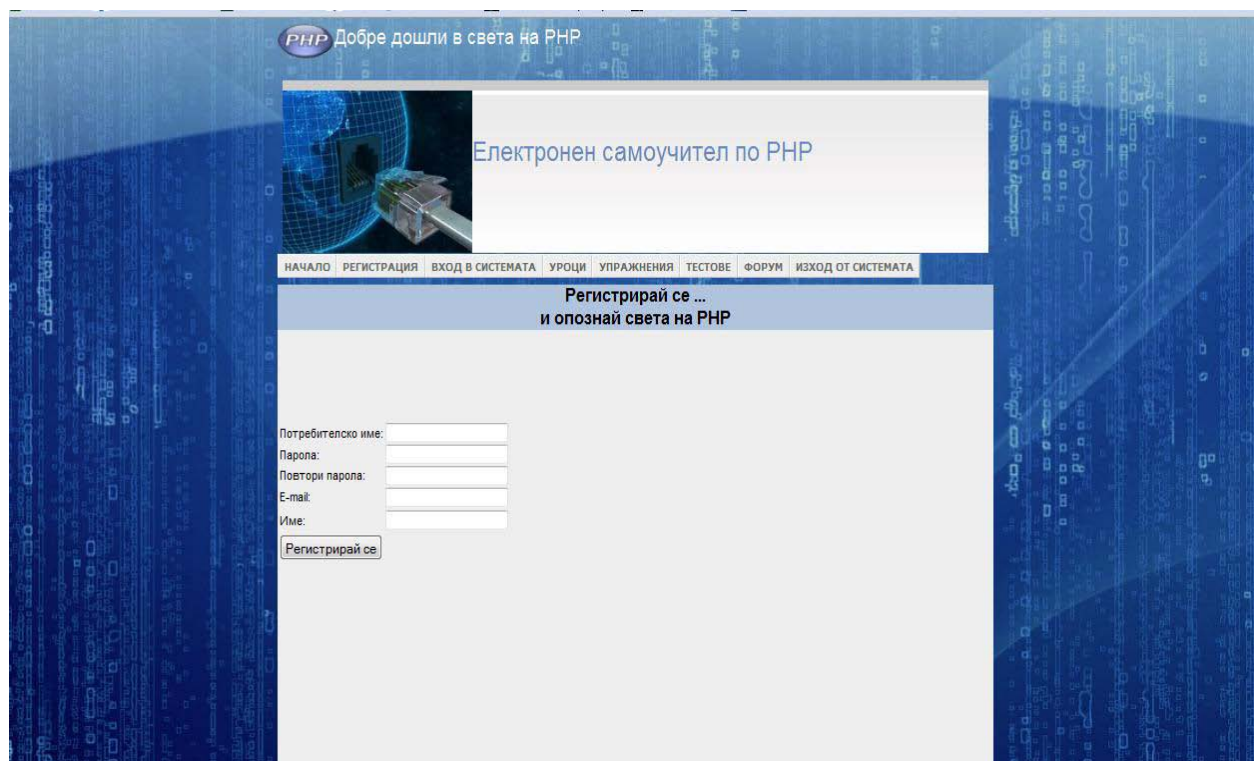
При стартиране на сайта от потребителя, се зарежда началната страница index.php с кратко описание на същността на уеб приложението.



Дипломна работа: „Електронен самоучител по РНР”

От навигационната лента потребителят може да избере, коя от основните страници на сайта да посети: „Начало”, „Регистрация”, „Вход в системата”, „Уроци”, „Уражнения”, „Тестове”, „Форум” и „Изход от системата”.

Страницата „Регистрация” позволява на потребителя да създаде своя собствена регистрация, която ще му позволи да използва цялостния капацитет на приложението.



РНР Добре дошли в света на РНР

Електронен самоучител по РНР

НАЧАЛО | РЕГИСТРАЦИЯ | ВХОД В СИСТЕМАТА | УРОЦИ | УПРАЖНЕНИЯ | ТЕСТОВЕ | ФОРУМ | ИЗХОД ОТ СИСТЕМАТА

Регистрирай се ...
и опознай света на РНР

Потребителско име:

Парола:

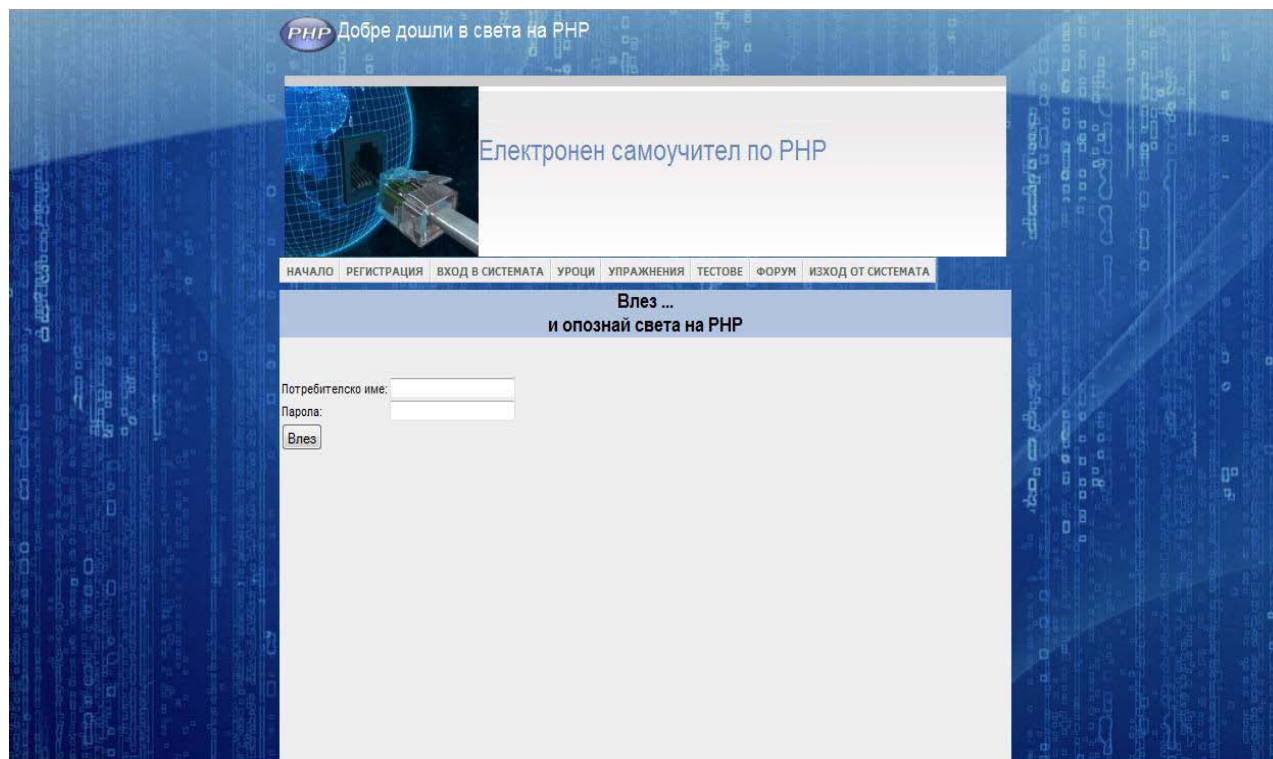
Повтори парола:

Е-мейл:

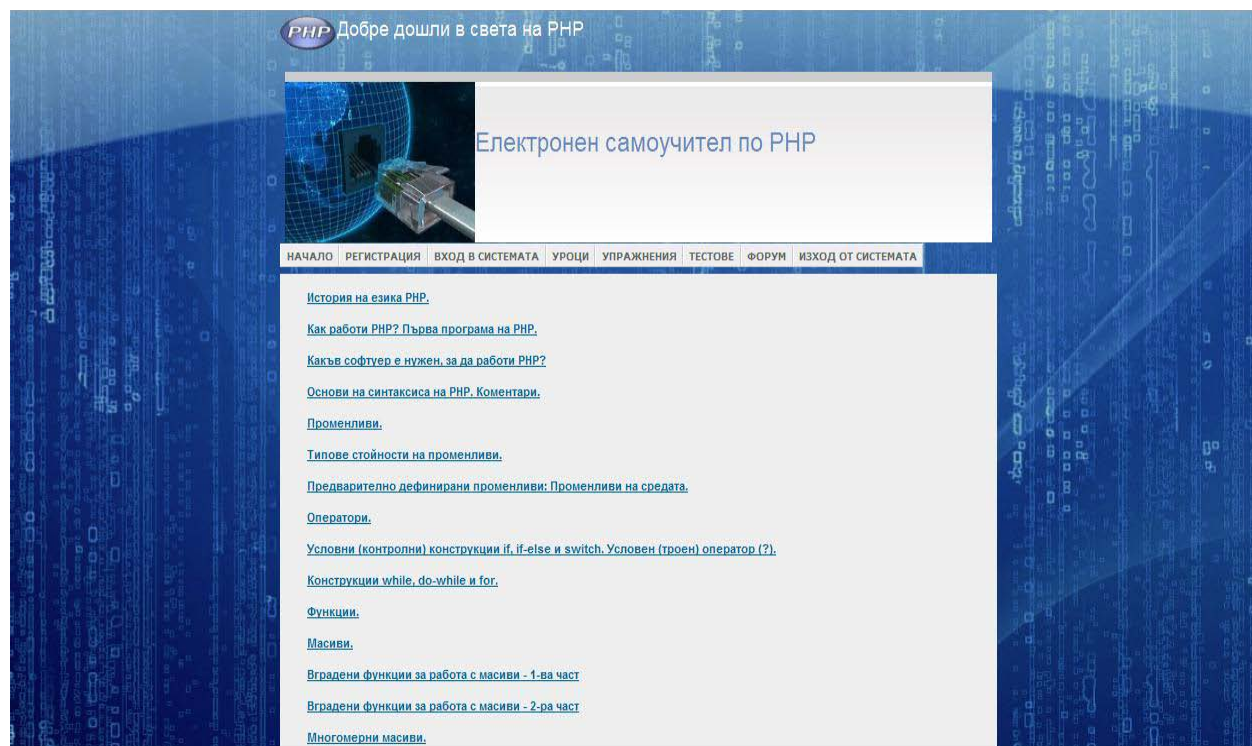
Име:

При избор на страницата „Вход в системата” се визуализира входната форма на уеб приложението.

Дипломна работа: „Електронен самоучител по PHP”



Страницата „Уроци” предлага на потребителя избор измежду 20 урока по PHP. Уроците са написани на достъпен език, илюстрирани със схеми и примери.



Дипломна работа: „Електронен самоучител по PHP”

The screenshot shows the 'Electron PHP Tutor' website. At the top, there is a navigation bar with links: НАЧАЛО, РЕГИСТРАЦИЯ, ВХОД В СИСТЕМАТА, УРОЦИ, УПРАЖНЕНИЯ, ТЕСТОВЕ, ФОРУМ, ИЗХОД ОТ СИСТЕМАТА. The main content area is titled 'Как работи PHP? Първа програма на PHP.' and contains text explaining the basics of PHP, HTML, and the role of the browser and server. Below the text is a diagram illustrating the workflow between a Server and a Client via the Internet. The diagram shows the following steps: 1. Demande de la page par le client (Client requests the page), 2. Chargement de la page sur le disque (Page loaded on the disk), 3. Traitement de la requête (Request processing), 4. L'interpréteur PHP génère du HTML (PHP interpreter generates HTML), 5. Réponse avec du code HTML (Response with HTML code), 6. Réponse au client (Response to the client). The background of the website has a blue, digital, circuit-like pattern.

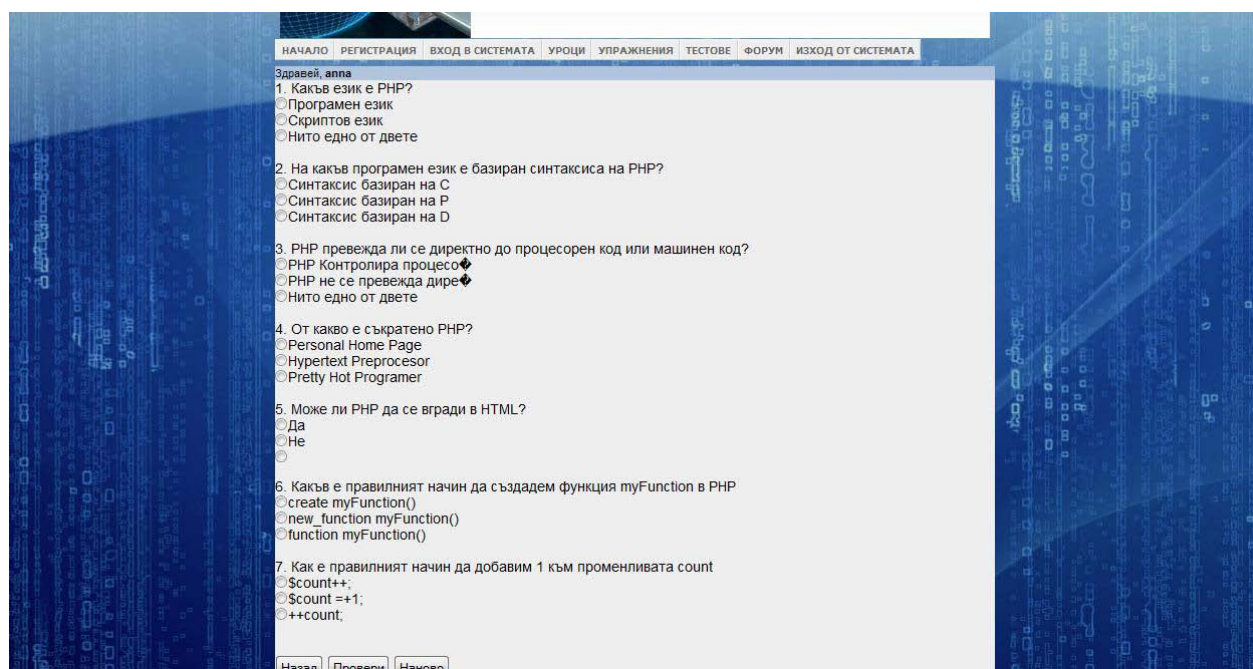
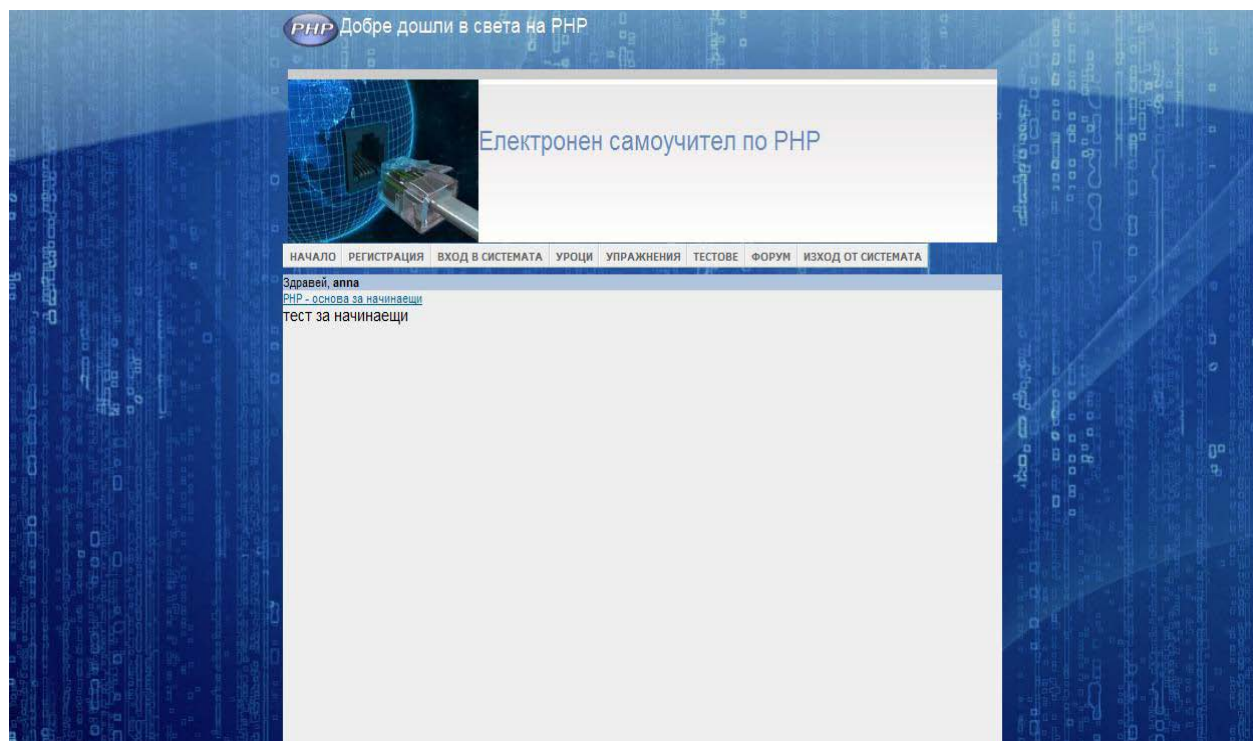
посетител отваря някаква php страница в интернет и браузърът изпраща заявка към сървъра. PHP

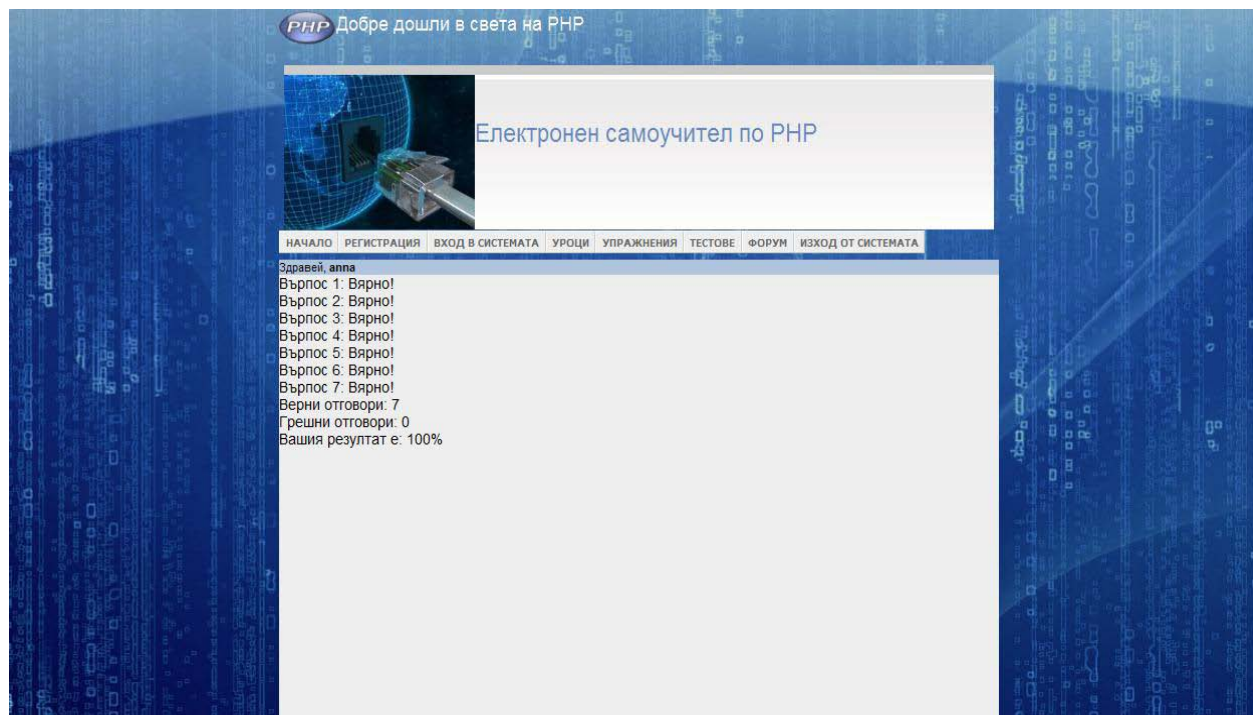
Страницата „Упражнения” предоставя възможност за избор измежду упражнения добавени от потребители на сайта, а ако разглеждащият е регистриран потребител – може да добави свое упражнение.

The screenshot shows the 'Electron PHP Tutor' website, specifically the 'Упражнения' (Exercises) section. The navigation bar is the same as in the previous screenshot. Below the navigation bar, there is a list of exercises with links: [Лесно създаване на шаблони в PHP](#), [Промяна на размера на текста чрез използване на PHP](#), [PHP: Колко посетители са online](#), [Обикновен скрипт за изпращане на Email чрез формуляр PHP](#), and [Урок за лесно странициране в PHP](#). Below the list, there is a button that says 'Тук можете да добавите ново упражнение :)))'. At the bottom, there is a form with a label 'Име:' and a text input field, and a label 'Съдържание:' with a larger text area. A 'Запиши' (Save) button is located at the bottom right of the form. The background of the website has a blue, digital, circuit-like pattern.

Дипломна работа: „Електронен самоучител по PHP”

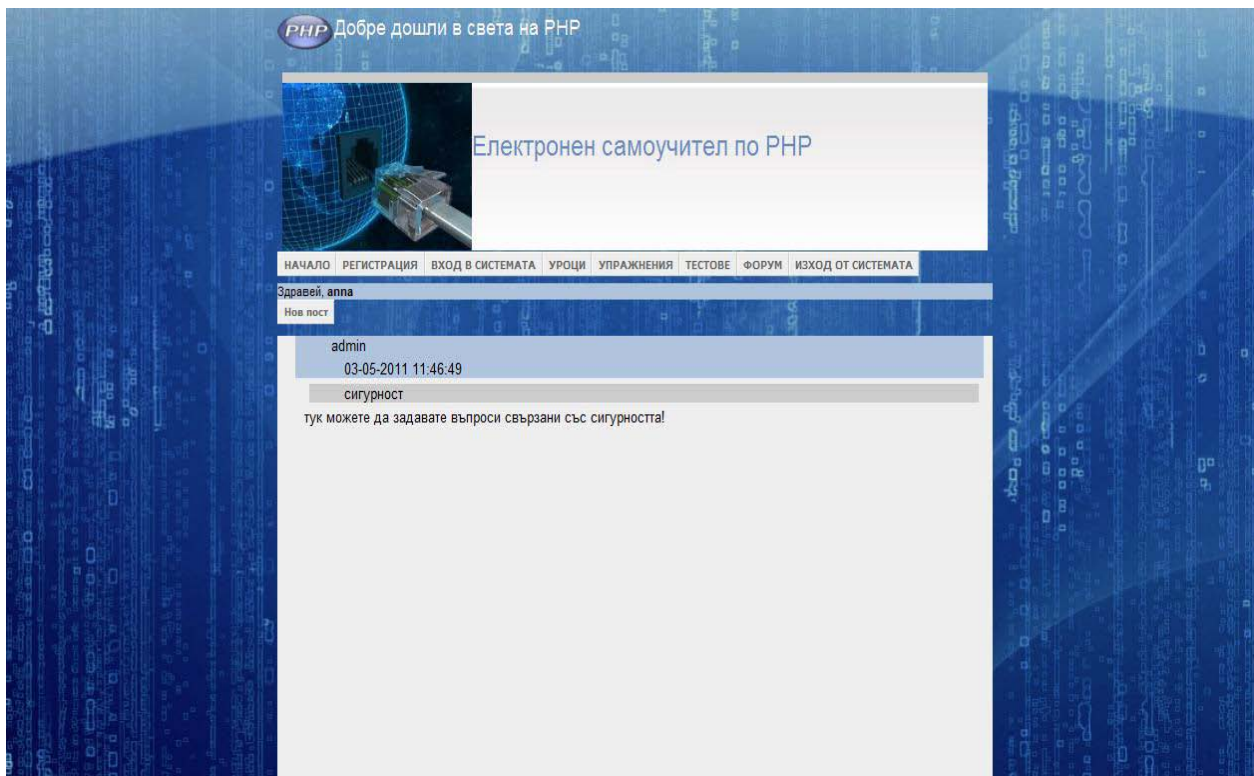
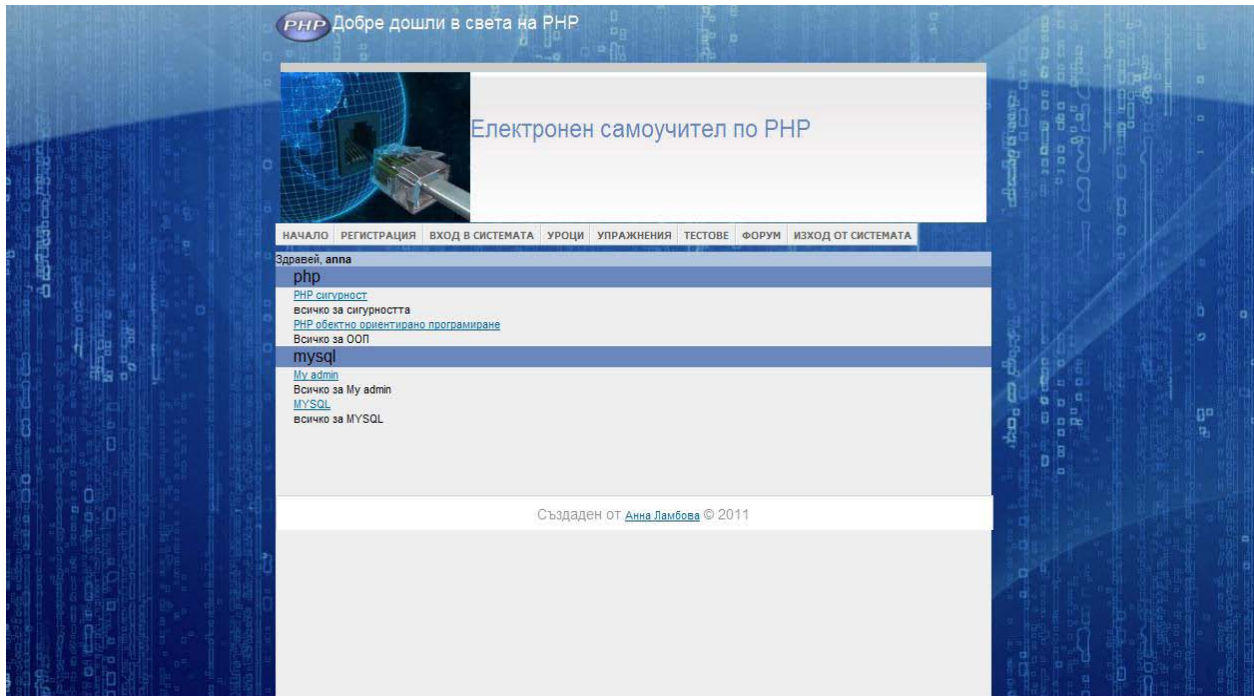
Страницата „Тестове” предоставя списък с налични тестове, кратко описание на теста, както и възможност за тяхното решаване и проверка.





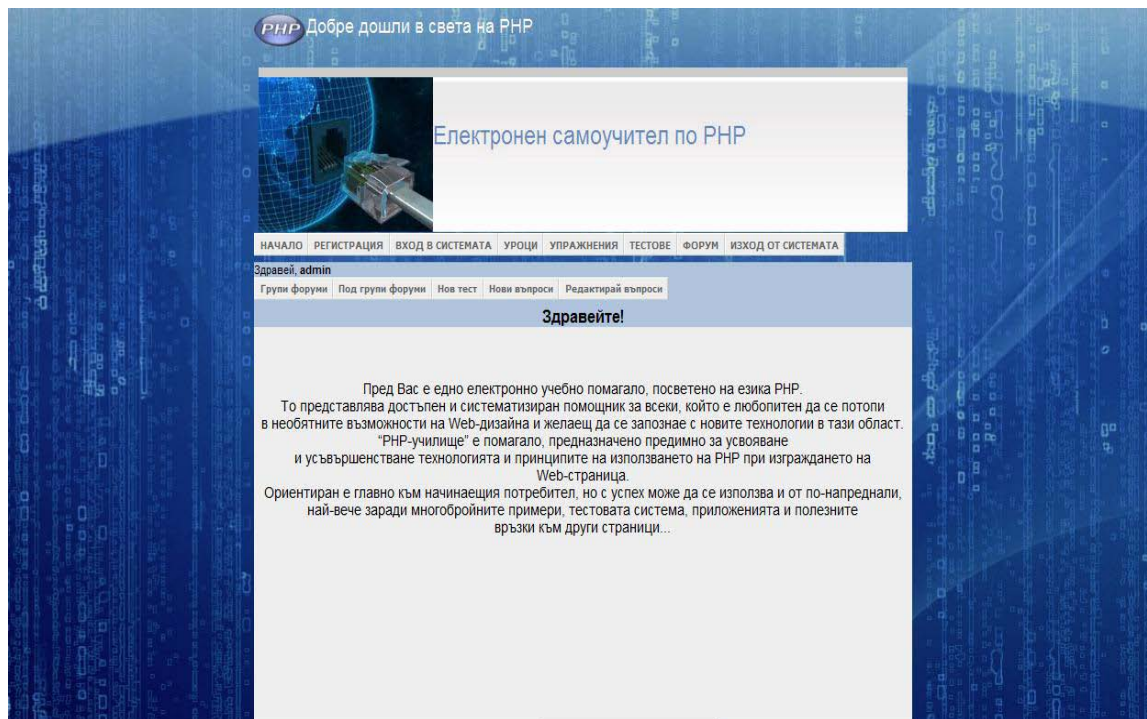
Страницата „Форум” съдържа списък с групите и активните подгрупи във форума, като при избор на конкретна група, пред потребителя се визуализира съдържанието на темата, а ако той е регистриран, му се предоставя възможност да изкаже собственото си мнение по въпроса.

Дипломна работа: „Електронен самоучител по PHP”



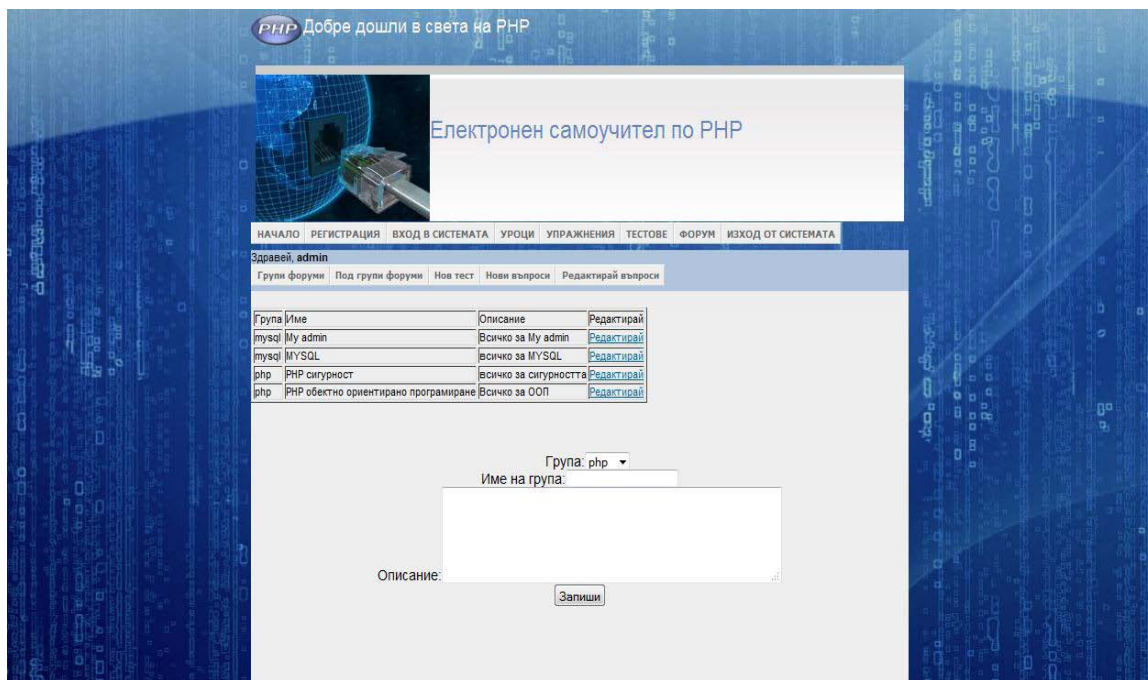
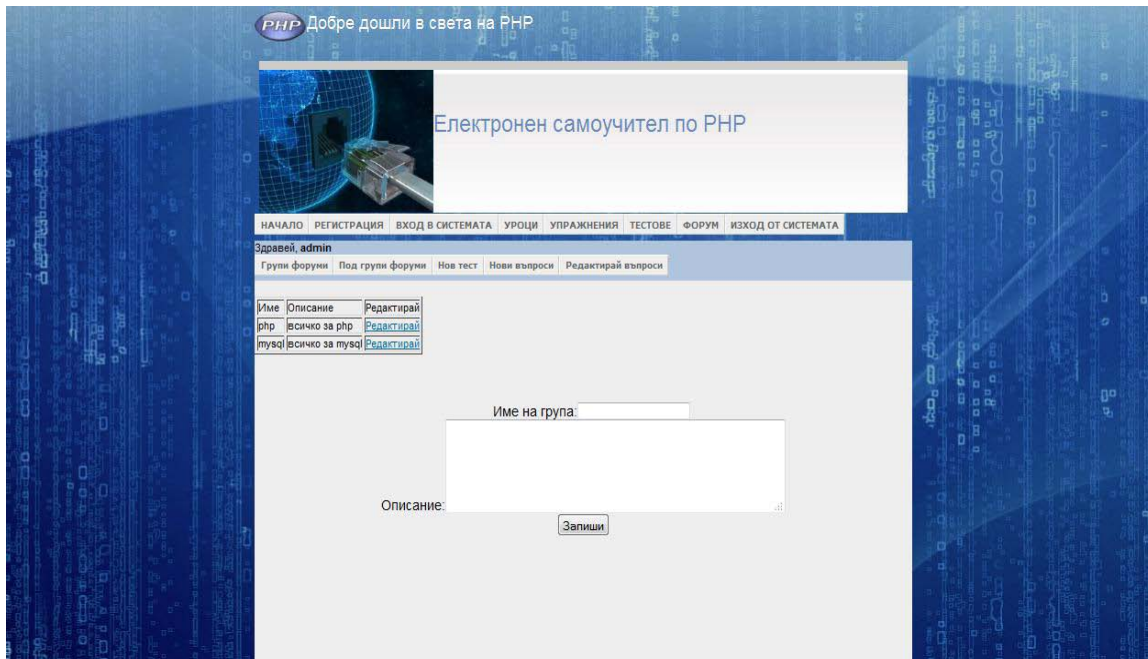
Дипломна работа: „Електронен самоучител по PHP”

Ако потребителя има администраторски права, пред него се визуализира допълнително – администраторско меню.



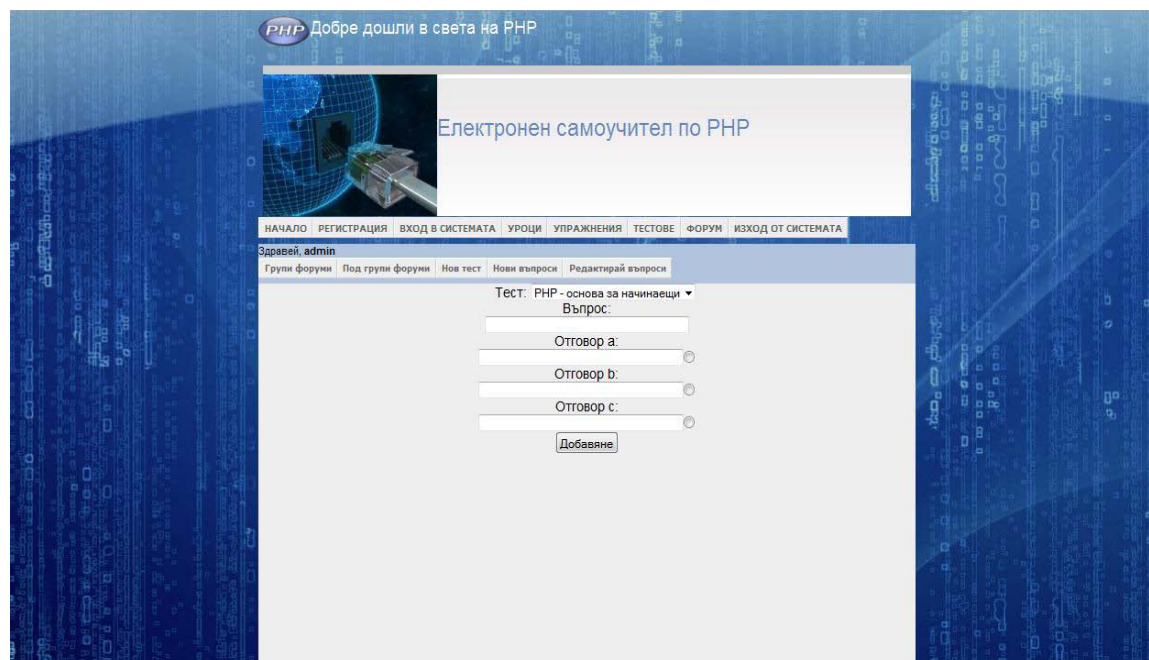
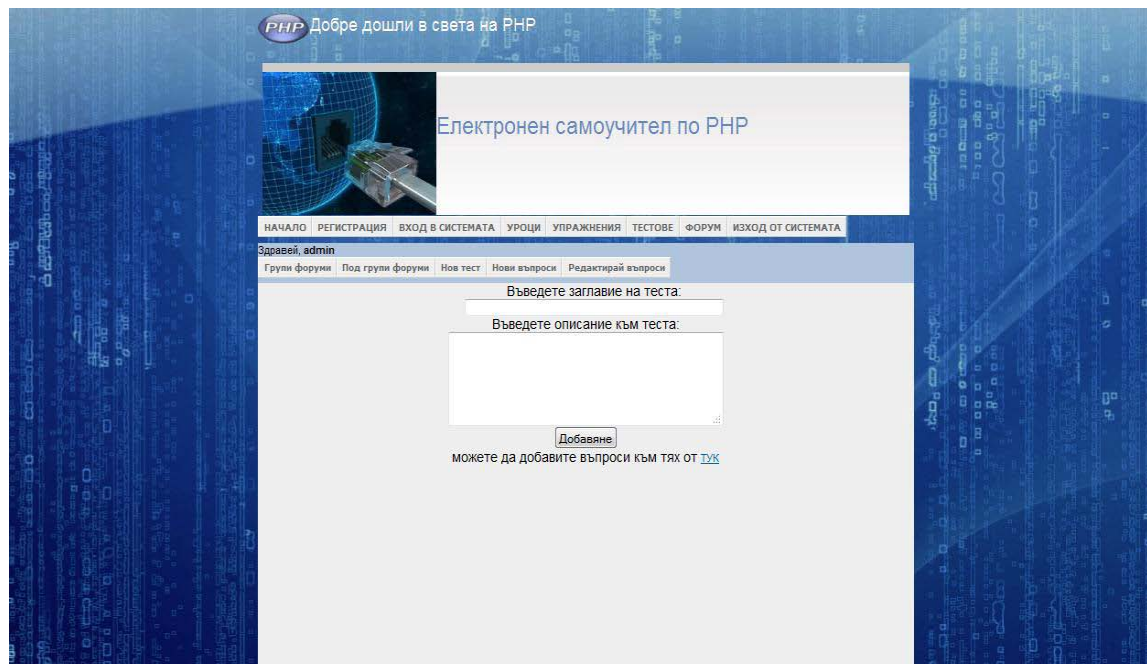
То позволява, на администратора, да създава и редактира групите и подгрупите на форума.

Дипломна работа: „Електронен самоучител по PHP”



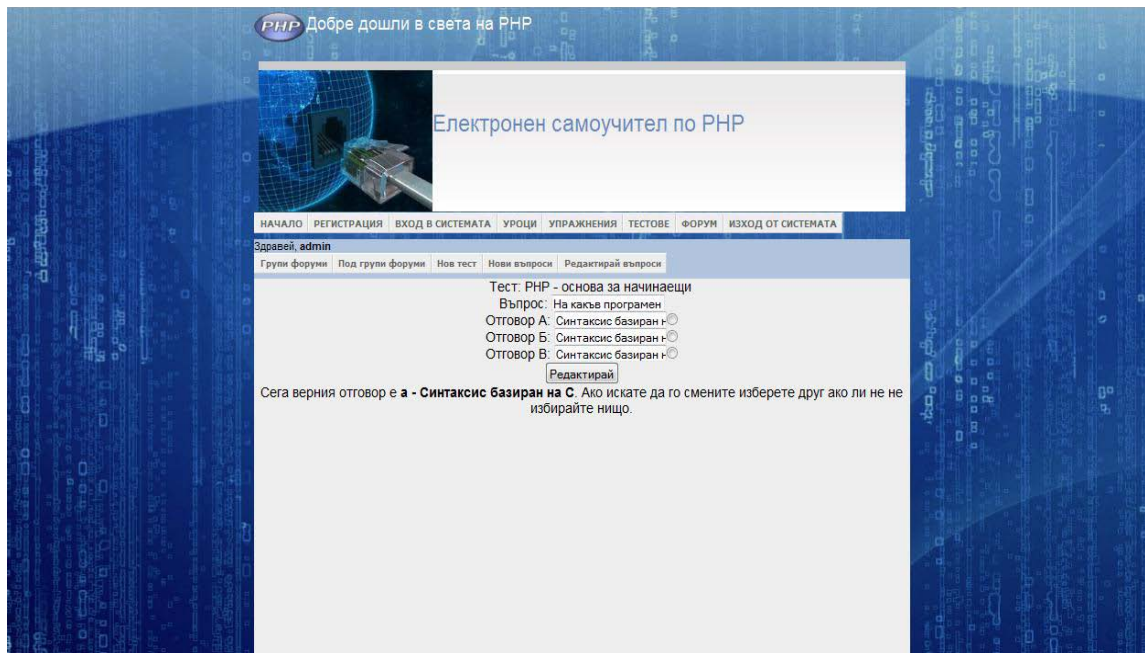
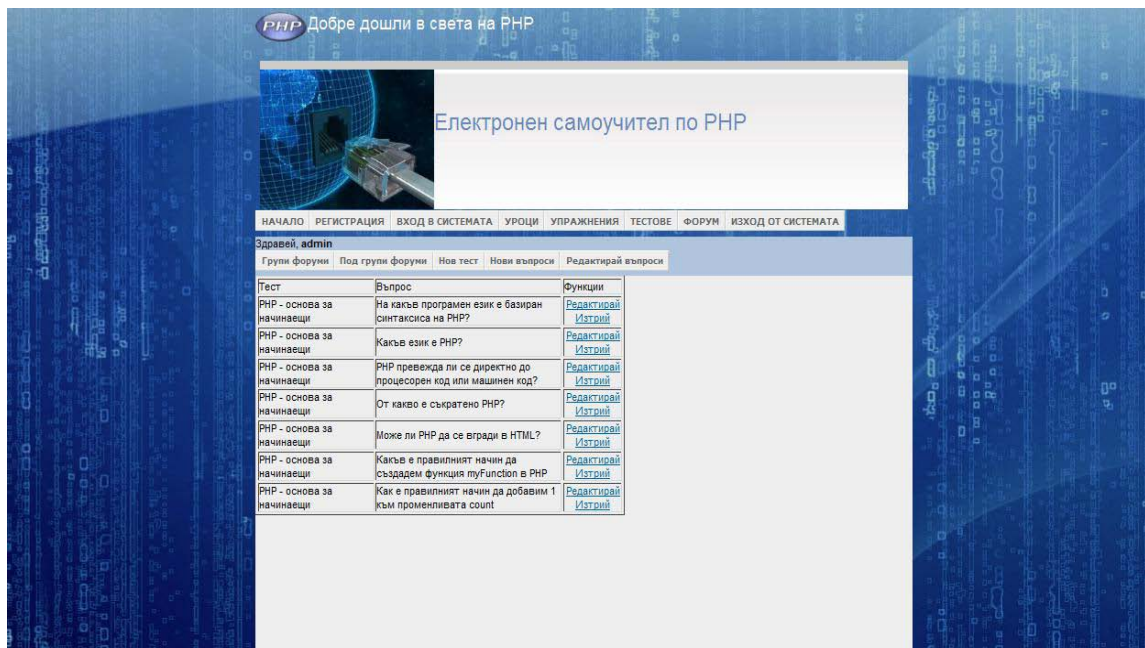
Дипломна работа: „Електронен самоучител по РНР”

Както и да създава нови тестове и да създава нови, и коригира стари въпроси от тестовете.



Дипломна работа: „Електронен самоучител по PHP”

В страницата „Редектирай въпрос”, се предоставя списък с всички въпроси и възможност за редактиране или изтриване на въпроса. Като при изтриване, се дава възможност за промяна на решението.



При избор на „Изход от системата” от менюто, потребителя се препраща към началната страница, като се анулира отворената сесия.

ГЛАВА 5. ЗАКЛЮЧЕНИЕ

Дипломната работа представлява вид информационна система, обслужваща обикновени потребители и администратори. WEB базираната система е реално функционираща и покрива нуждите на потребителите.

Информацията е отсята внимателно, поднесена е чрез разнообразни примери като се акцентира на най – важните моменти от процеса на обучението по РНР. Сайтът се превръща в един своеобразен самоучител, който би бил от полза както на учениците, така и на студентите в курса им на обучение по РНР.

Разбира се в приложението има какво още да се доусъвършенства, с напредването на науката и техниката ще се появи нуждата от по-голяма интерактивност , по – голям достъп до учебния материал, и т. н. Смятам, че материята на сайта ще благоприятства подобни нововъведения - една от основните ми идеи е да бъде отворен за подобни проекти. Да позволява да бъде променян в бъдеще. Да е достъпен за новостите, а не да ги възпрепятства.

ЛИТЕРАТУРА

1.Програмиране и WEB дизайн с PHP5, Apache, MySQL – Том 1 и Том 2.
АлексСофт 2005

2.Бързо и ефективно създаване на WEB страници. Софтпрес 2004

3.Уеб приложения за бази данни с PHP и MySQL –O'Reilly 2003

4.Ръководство на програмиста SQL . Софтпрес 2001

5.Създаване на WEB страници – учебен курс Софтпрес 2003

6.JavaScript – ръководство на програмиста . Софтпрес 2001