|  |
| --- |
| **ПЛОВДИВСКИ УНИВЕРСИТЕТ „ПАИСИЙ ХИЛЕНДАРСКИ”**  Факултет по математика и информатика  Катедра: „Компютърна Информатика”  Специалност: „Бизнес информатика с английски език”  Степен: ”магистър”  ДИПЛОМНА РАБОТА  на тема:  **„Интернет портал за технологични новини”**  Дипломант: Научен ръководител:  Владимир Бакалов гл. ас. Светослав Енков  фак. № 1001437010      Пловдив 2012 г. |

Увод

През последните години Интернет се наложи като един от основните начини за научаване на новини от всички сфери на обществения живот. В Мрежата може да бъде намерена информация на всякакви теми – от здраве и култура, през различните области на науката, та чак до самите устои на Интернет – информационните технологии. Всъщност, исторически погледнато, Интернет технологиите са възникнали като начин едни учени от едни университети да си разменят научни трудове и статии (включително и на ИТ тематика) с други учени от други университети. И днес, повечето хора, когато чуят за интернет, се сещат първо за новини от сферата на информационните технологии.

Аз, като човек, който живо се интересува от информационните технологии и положителното им влияние върху живота ни, реших да създам уеб портал, съдържащ новини и ревюта на компютри, хардуерни компоненти и софтуер, който да се пълни ежедневно с интересни и полезни новини. Сайтът ми е предназначен не само за ИТ ентусиастите, но и за всички хора, които искат да бъдат в крак с новите тенденции в сферата на компютрите. Постарал съм се сайта да е добре организиран и подреден, за да може всеки да намира бързо и лесно желаната от него информация.

В настоящата дипломна работа описвам как създавам портала. Ще се запознаем подробно с принципите, техниките и инструментите, използвани за създаването на сайта, както и с взаимовръзките между тях.

Адресът на уеб сайта е [www.bakalov.cloudvps.bg](http://www.bakalov.cloudvps.bg)

Необходими предпоставки за осъществяване на поставената цел:

1. Back end

a) Създаване на база данни, която да пази информация за новините, ревютата, категориите и цялата останала информация, свързана със сайта - реализира се с MySQL.

б) Използване на скриптов език PHP за програмиране от страна на сървъра. PHP реализира вътрешната програмна логика на сайта и осъществява връзката с базата данни.

в.) Използване на Wordpress CMS (content management system) за управление на съдържанието.

2. Front end

а) Създаване на модерен потребителски интерфейс, използвайки най-новите стандарти за писане на валиден HTML и CSS.

Глава 1

Основни характеристики на уеб приложенията

## 1. История на Интернет

Световната електронна мрежа Интернет е създадена в САЩ към 1960г. в отговор на изстрелването на първия спътник в света - съветския "Спутник1". Корпорацията RAND, осигуряваща американската военна машина, разработва комуникационна мрежа, която да действа дори след ядрен удар, т.е. ако един компютър от мрежата бъде унищожен, това да не води до прекъсване на връзките. При централен компютър в мрежата, твърде вероятно е противникът да го избере за първа цел. Корпорацията RAND предлага принципно нов вид мрежа без централен компютър. Всичките й възли да са с равноправен статут. Всеки от тях да може да приема и изпраща съобщения.

Освен принципа за децентрализация, друг важен и основополагащ принцип, прилаган след 1960г., е разпределянето на предаваните данни в пакети. Така мрежата става пакетно-ориентирана.

Важен атрибут на един пакет е неговия адрес. Ако съобщението е голямо, то се разделя на няколко пакета. Тези пакети започват своя път от някакъв възел и го завършват в друг. За пакета се грижат специални програми, които обикновено работят на компютри, наречени маршрутизатори или рутери. За всеки пакет се проверява къде трябва да отиде и чрез специални таблици се насочва към друг маршрутизатор. Пакетите пътуват директно само между два най-близки възела от мрежата. В останалите случаи се извършва неколкократно прехвърляне, докато се стигне до целта.

Пътят, по който се движи пакетът, не се счита за важен. Така, ако един или няколко възела в мрежата престанат да функционират, пакетът ще бъде предаден, стига да има маршрут от началната до крайната точка.

Първата опитна мрежа, построена върху горните принципи, е била съоръжена във Великобритания през 1968 г. Скоро след това Пентагонът чрез агенцията ARPA (Advanced Research Projects Agency) започва да изгражда подобна мрежа в САЩ. През 1971 г. тази мрежа, наречена ARPANET, има вече 23 възела, повечето от които са бързи компютри, намиращи се в университетски центрове. Тогава едно от важните приложения на мрежата е било, че е могло разумно да се разпределя и използва скъпо струващото машинно време и оперативна памет, която се е ползвала от големи разстояния. Днес смисълът за това трудно може да се разбере, защото и обикновеният персонален компютър има 8 Гигабайта RAM. Но през 1971 г. главният компютър, например на Университета в щата Юта, е имал общо 12 килобайта RAM.

Годината 1972 е забележителна с това, че тогава е изобретена първата компютърна програма за изпращане и получаване на електронна поща (e-mail) и оттогава започва да се измества основното предназначение на компютърните мрежи: учените все повече ги използват за комуникации помежду си, вместо за изчисления от разстояние. Следващата година се появяват и първите възли на ARPANET, намиращи се извън САЩ -- в Англия и Норвегия. Бързият ръст на мрежата се определя и от факта, че към нея могат да се включват компютри от различни типове и марки. През 1974 г. е публикуван и първият стандарт за протокол за пакетна връзка (TCP). През следващите години се появява стандарт UUCP (Unix to Unix Copy) и през 1979 г. -- основаната на него система за обмен на електронни новини USENET, представляваща в известен смисъл развитие на електронната поща.

На 1 януари 1983 г. се въвежда стандартът за протокол TCP/IP за мрежата ARPANET и се появява названието Интернет. Тогава от ARPANET се отделя сегментът MILNET, предназначен за военните служби в САЩ.

През 1984 г. растежът на мрежата набира сили, като броят на включените компютри надминава хиляда. По това време се въвежда и принцип за наименуване на отделните възли, известен като DNS (Domain Name System). През 1989 г. включените компютри стават 100 000, а през 1992 -- надминават 1 000 000.

През 1993г. започва епохата на световната паяжина (WWW -World Wide Web) или Уеб, което на български се превежда като "Световно разпространена мрежа". Тя сега е това, за което много хора мислят, когато казват "Интернет", въпреки че всъщност е само част от него.

## 2. Отличителни черти на световната компютърна мрежа Интернет

Интернет е световно разпространена съвкупност от мрежи, шлюзове, сървъри и компютри, използващи общ набор от телекомуникационни протоколи за връзка помежду си. Интернет е построена като мрежа, обединяваща в себе си голямо количество от съществуващите мрежи - локални, регионални и глобални. Разликата между глобалната мрежа и Интернет е в използваните протоколи за връзка и за предаване на данни. Всяка глобална мрежа може да бъде част от Интернет, ако използва семейство протоколи TCP/IP.

Структурата на Интернет е построена като мрежа, състояща се от множество под-мрежи. В нейната структура се отделя магистралната мрежа, а мрежите присъединени към нея се разглеждат като системни. Магистралната мрежа и всяка от автономните системни мрежи имат собствено административно управление и собствени протоколи за маршрутизация. Автономните системи не се свързват една с друга, а използват магистрална мрежа. Смисълът на разделянето на цялата мрежа Интернет на под-мрежи е да бъде възможно разширяването й.

## 3. Термини при работа в Интернет

**- *Домейн* -** името му еднозначно определя и посочва свързан към Интернет компютър (хост) или услуга. Често то има нещо общо с името на дадена фирма, организация или услуга и подлежи на регистриране почти по същия начин, по който се регистрират и запазените марки. Наред с името на домейн адресът може да съдържа и името на под-домейн, типа на домейна и код за страната. В момента се срещат домейни главно от следните типове: ас (академичен), com и со (фирмен или комерсиален), edu (образователен), gov (правителствен), mil (военен), net (административен), org (нестопанска организация). За всяка страна е определен отделен код, макар и той не винаги да се използва, например: us (САЩ), au (Австралия), ca (Канада), de (Германия), es (Испания), fr (Франция), uk (Великобритания).

**- *IP-адрес* -** IP адресът е уникален номер, много наподобяващ телефонен номер, който се използва от машини (обикновено компютри), за да се свързват едни с други, когато изпращат информация през Интернет, използвайки Интернет протокол (IP). Той позволява на машините, които предават информацията, да знаят къде да я изпращат, а на машините, които получават информацията, да знаят, че тя идва от желаното местоназначение.

Да вземем за пример IP адреса 207.142.131.236. Получаването на тези числа от домейн адреси, написани в една по-разбираема за човека форма, като например www.wikipedia.org, се извършва от Система за имена на домейни (DNS). Този процес е известен като преобразуване на имената на домейни към АйПи адрес. (Забележка: Система за имена на домейни е общо понятие, обхващащо всичките Сървъри за имена на домейни)

**- *Пропусквателна способност* -** измерва се в битове за секунда (bps) и характеризира капацитета на връзката към Интернет. От стойността й зависи с каква скорост се изтегля информация от Интернет.

**- *World Wide Web* -** или просто **Web** (букв. световна паяжина), е хипертекстова система за обмен на мултимедийна информация, съхранявана на компютри, влизащи в състава на Интернет. Тези компютри се наричат мрежови сървъри (уебсървъри, web servers). Под мултимедийна информация се разбира съвкупност от текст, графични изображения, видеоклипове, звукозапис и анимация. В това отношение това е най-добрата услуга в Интернет.

Гръбнакът на WWW е използваният за пренос мрежов протокол HTTP (на английски: HyperText Transfer Protocol, за прехвърляне на хипертекст). HTTP e протокол от ниво 7 (според модела на OSI), което го прави изключително гъвкав за използване. Версията HTTPS използва криптиране (шифриране) с помощта на SSL или TLS протоколите, което го прави незаменим при WWW-транзакции, изискващи повишено ниво на сигурност.

**- ISO/OSI model** (Internetional Standarts Organization / Open System Interconnect) Това е съвкупността от всички правила и модели за работа,изпълнение на алгоритмите и всички функции на услугите в Интернет.

**- “Протокол” (protocol)** Правила, дефиниции и стандарти за обмен на данни по даден комуникационен канал.

**- "Информационна система"** ( “IS, information system, cluster” ) е устройство или система от свързани устройства, които или някое от които е предназначено да съхранява, обработва, изпраща или получава цифрова информация или преобразувана в такава ,информация от различен характер.

**- "ПОТРЕБИТЕЛ" (“client”)** e физическо лице, софтуер, техническо устройство или всички взети заедно, което или които след извършена регистрация ползва предоставени му Интернет услуги съгласно установени правила - (протоколи).

**- “ДОСТАВЧИК” (“Internet provider”)** е физическо лице, софтуер, техническо устройство или всички взети заедно, осигуряващо или осигуряващи на ПОТРЕБИТЕЛЯ различни Интернет услуги съгласно установени правила - (протоколи).

**- "Клиентски профил на ПОТРЕБИТЕЛЯ"** (user profile) съдържа информацията за ПОТРЕБИТЕЛЯ, предоставяна в процеса на регистрация при сключване на договора и съхранявана на сървъра на ДОСТАВЧИКА.

Достъп до клиентският профил на ПОТРЕБИТЕЛЯТ имат само ДОСТАВЧИКА и ПОТРЕБИТЕЛЯ. Достъпът от страна на ПОТРЕБИТЕЛЯ се осъществява чрез въвеждане на потребителско име и парола.

**- "Злоумишлени действия" (“crash”, “hacking”)** са действия или бездействия, които могат да се квалифицират като престъпление или административно нарушение по българското законодателство или по приложимото право на друга страна, както и действия или бездействия, които нарушават Интернет етиката, добрите нрави или нанасят вреди на лица, свързани към Интернет или асоциирани мрежи. Злоумишлени са следните неизчерпателно изброени действия: изпращане на нежелана поща

**- spam, junk mail** - препълване на каналите чрез изпращане на нежелани IP пакети с цел генериране на нежелан трафик (flood), получаване на достъп до ресурси с чужди права и пароли, използване на недостатъци в системите с цел собствена облага или добиване на информация (hack), извършване на действия, които могат да бъдат квалифицирани като промишлен шпионаж или саботаж, повреждане или разрушаване на системи или информационни масиви (crack), изпращане на "троянски коне" (Trojan horses), предизвикване инсталация на вируси (viruses), системи за отдалечен контрол и за следене на чужда информация (sniff), смущаване нормалната работа на останалите потребители на Интернет и асоциираните мрежи, други подобни.

**- "Сървър" ("server")** е устройство, система от свързани устройства или софтуер, на което или на някое от които е инсталиран системен софтуер (сървърен процес или демон) за изпълняване на задачи във връзка със съхраняване, обработка, приемане или предаване на информация.

**- "Хипервръзка" ("hyperlink")** е изведен и визуализиран на съответната Интернет страница, интернет адрес, улесняващ пренасочването на ПОТРЕБИТЕЛЯТ към други Интернет страници на същия уебсайт или към други уебсайтове или ресурси в глобалната Интернет мрежа, чрез натискане ("кликване") върху него.

**- “Прокси сървър” ( “proxy server”)** Такъв сървър, който се използва за събирането (кеширане) на информацията и обръщенията към него ускоряват достъпът - (бързодействието) до интернет ресурси.

**- Електронната поща,** също имейл (от англ.e-mail или email), е начин за създаване, съхраняване, изпращане, приемане и препращане на съобщения посредством електронни комуникационни системи. Терминът се отнася за всички „публични“ (общодостъпни) интернет e-mail системи, базирани на протокола SMTP, както и за „частните“ (с ограничен достъп) интранет системи, които доста често използват интернет протоколи за вътрешните си e-mail услуги.

**Основни възможности, които предлагат програмите за електронна поща:**

* Изпращате и получавате текстови съобщения. Можете да изпратите имейл на всеки, който има имейл адрес. Съобщенията пристигат във входящата кутия за електронна поща на получателя за секунди или минути – независимо дали се намира в съседната сграда или на другия край на света.
* Електронната поща е двупосочна. Можете да получавате съобщения от всеки, който знае вашия имейл адрес, а след като ги прочетете, да им отговаряте.
* Изпращате и получавате файлове. Освен текст, можете да изпращате с имейлите всякакви видове файлове, включително документи, картини и музика. Файл, изпратен в имейл съобщение, са нарича прикачен файл.
* Изпращате съобщения до група хора. Можете да изпратите едно имейл съобщение едновременно до много хора. Получателите могат да отговарят на цялата група, което позволява групови обсъждания.
* Препращате съобщения. Когато получите имейл, можете да го препратите на друг, без да му отговаряте.
* Едно от предимствата на електронната поща пред телефона или обикновената поща е нейното удобство: Можете да изпращате съобщения по всяко време на деня и нощта. Ако получателите не са пред своите компютри и онлайн (свързани с Интернет), когато изпращате съобщението, те ще го намерят следващия път, когато проверяват пощата си. Ако са онлайн, можете да получите отговор след минути.

ГЛАВА 2

Същност, техники и инструменти за разработване на Интернет приложения

Много Интернет приложения съдържат статично съдържание, като научни доклади и статии. Страниците на тези приложения представляват документи, съдържащи обикновен текст, изображения, препратки към други документи. Най-често използваните средства за разработване на такива интернет-сайтове са HTML (HyperText Markup Language) и CSS (Cascading Style Sheets), които предоставят различни методи за структуриране и представяне на съдържанието на страницата

## 1. Уеб-сървъри

 Сървърът представлява компютър, доставящ една или повече услуги през компютърна мрежа, обикновено през request-response програма. Тези сървъри се поддържат от определени приложения, които представляват компютърни програми направени с цел да се справят с много и различни заявки (requests). Примери за подобни приложения са Интернет пощенските сървъри, сървъри за файлове, уеб сървъри и прокси сървъри.

Сървър може да бъде и компютър отделен от другите за да изпълнява определена работа от специализирано приложение. Например когато софтуерът Apache HTTP Server се използва като уеб сървър за сайт на хостинг компанията, компютърът използващ програмата също се нарича сървър. Приложенията могат да бъдат разпределени по сървърите на големи разстояния, според това колко са натоварени.

Сървър също се използва и като обозначение на компютърни модели предназначени за управление на приложения с голям капацитет на работа, наричани още оперативни единици, често работещи сами за големи периоди от време. Докато всеки обикновен компютър може да бъде сървър, специализираните такива обикновено изискват подходящ хардуер. Те включват по-бърз процесор, бърза и голяма като размер RAM памет, както и по-големи и бързи твърди дискове. По-очевидните белези са голямото потребление на електричество, мрежовите връзки, кибернетични запомнящи устройства както и променливата конструкция на така наречените Blade сървъри, които се използват при подобни машини.

## 

## 2. Страници и уеб-браузъри

Когато даден потребител посещава интернет-сайт, той гледа страниците на сайта, които съдържат текста, графиките, звука и видеото. Въпреки че една интернет-страница не е със същите размери или формат като разпечатана страница, думата "страница" се използва, за да се направи разлика между страници, папки и сайтове. Повечето уеб-сървъри са настроени да търсят автоматично страница, наречена "index" като основна за всяка папка.

Интернет-браузърите представляват софтуерни програми, които позволяват да се виждат интернет-страници. Интернет-браузърите интерпретират HTML кода и предоставят визуално разположение на екрана.

## 3. Статични и динамични интернет-сайтове

### 3.1. Статични интернет-сайтове

Докато милиони хора все още разчитат на добрите стари черно-бели вестници всеки ден, те ще са нашият перфектен пример за статични сайтове. След като е отпечатан вестника вече не може да бъде променен, вида му е определен и просто ще се наложи да изчакате докато се появи следващия брой. Представете си обаче, че най-добрия ви репортер току-що е донесъл разбиваща, сензационна новина, а вие не можете да я публикувате, защото вече е късно.

По същия начин, след като статичен уеб сайт е вече публикуван, той няма да се промени твърдо до следващото издание; трябва уебмастъра, или някой който е запознат с HTML, флаш или редактиране на това специфично съдържание да се заеме и да се потруди здраво.

Първото поколение е поколението на статичното публикуване - страници, които разчитат на HTML, статични изображения, тестове, които не могат да се позиционират точно от гледна точка на X и Y координатите. Такива страници са доста обикновени. За да се получат наистина задоволяващи резултати от тях чрез техники, трябва разработчика да е експерт в HTML или да е графичен дизайнер. За да промени страницата, той ще трябва да коригира HTML ръчно или с редактор. Освен това статичните страници не са съвместими с базите данни. Те не могат да правят нещо много по-различно от това да показват текст или изображения.

От известно време Мрежата се ориентира към активни интернет-сайтове, които позволяват на потребителя да бъдат изпращани настроени според собствените му изисквания страници и предлагат по-динамичен опит при работа с браузъри. Такива сайтове се създават с помощта на комбинация от програмни езици и технологии, като може да се използва всяка от тези технологии поотделно или няколко заедно. Всички те са независими Тези технологии могат да бъдат разделени на две групи: технологии от страна на клиента и технологии от страна на сървъра. Първите включват: ActiveX контроли, създадени чрез Visual C++ или Visual Basic, Java аплети, скриптове от страна на клиента и динамичен HTML. Технологиите от страна на сървъра включват: CGI, патентовани APIs за уеб-сървър, ASP, JavaServer Pages и PHP.

**Положителни страни:** Статичният сайт позволява практически пълна свобода при създаването на дизайн, съобразен с всякакви изисквания от страна на клиента.

\* Статичният сайт позволява по-добро оптимизиране за търсещи машини на всяка отделна страница, което може драстично да подобри класирането на сайта в Google.

\* Поради липса на сървърна обработка и връзка с база данни, една статична страница се зарежда съществено по-бързо от същата по обем динамична, като това съотношение може да достигне до 60%.

\* Тъй като статичният сайт обикновено се изгражда и актуализира на локален компютър и едва след това се "качва" на сървъра, собственикът на сайта по всяко време разполага с актуално резервно копие от своя сайт. Това обикновено е изключително важно при възстановяване на сайта в случаи на сривове в сървъра или пробив в сигурността му.

**Отрицателни страни:** Статичният сайт се актуализира по-трудно и това обикновено е добре да се прави от самите уеб дизайнери.

\* При статични сайтове е по-трудно (понякога и невъзможно) да се създават по-сложни функционалности от типа на сортирания, филтрирания и т.н.

**Статичен сайт е по-подходящ за вас ако:**

Целта на сайта Ви е преди всичко онлайн представяне на Вашата фирма и ще включва неголям брой страници (до 20 - 30).

\* Не възнамерявате много често (ежедневно или ежеседмично) да актуализирате информацията в сайта.

\* Съзнавате, че всеки трябва да върши своята работа, и Вашата работа, като професионалист във Вашата област, не е да актуализирате сайтове.

### 3.2. Динамични технологии

Динамичния уебсайт, от друга страна използва програмиране в допълнение към оформлението и не само за да позволи на потока от данни в и извън обекта, но и за да направи смислени и бързи връзки с базата данни.

Така например онлайн бизнес указател може да позволи на местните фирми да се регистрират и да създадат профили за техните фирми. Посетителите на сайта може да търсят директно в директорията на базата данни според техните нужди, например един може да извършва търсене, за адвокат в техния пощенски код които владее испански или за денс училище в областта, която предлага танци курсове.

Динамичните уеб сайтове дават възможност техните собственици да качват информация през защитен с парола административен интерфейс.Не се изисква познания за HTML или дизайн на уеб сайтове, за да можете да обновите или поддържате такива сайтове. Още повече че всяка страница която създавате с подобна система изработена от нас е с визуален редактор за всяка страница, която правите и той прилича страшно много на Word на външен вид и по лесната си употреба. Изработка динамични сайтове.

Основната пречка пред вграждането на функционалност в клиентската част е, че създателят на системата няма контрол върху софтуера, който се използва за разглеждане на страницата. Тъй като всяка компания иска да обхване колкото е възможно повече потребители, използващи колкото е възможно повече различни браузъри, вграждането на технологии, които се поддържат само от новите версии на основните браузъри, е много бавно. Точно обратното, технологиите от страна на сървъра обикновено не изискват определен браузър и следователно се вграждат по-бързо.

- ActiveX контроли -

- Java аплети -

- Скриптове от страна на сървъра и DHTML –

\* Положителни страни: Динамичните сайтове са снабдени със система за управление на съдържанието, която позволява на обучено лице от фирмата-собственик да актуализира и допълва съдържанието на сайта, без да се налага за всяко нещо да се викат уеб дизайнерите.

\* Динамичните сайтове позволяват създаването на повече активни функционалности от типа на сортирания на продукти, търсения и филтрирания по даден признак и т.н.

\* Отрицателни страни: Динамичните сайтове като цяло по-трудно се поддават на детайлна оптимизация за търсещи машини и при равни други условия един динамичен сайт обикновено се класира по-зле в търсачките в сравнение с добре оптимизиран статичен сайт. Някои системи за управление на динамични сайтове (CMS) дори се опитват (неособено успешно) да имитират поведението на статичен сайт, за да "се харесат" на търсачките.

\* Тъй като при динамичния сайт всяка страница се генерира "по заявка", това отнема повече време на сървъра и динамична страница като правило се зарежда по-бавно в сравнение със статична страница със същото съдържание.

\* Тъй като при динамичния сайт съдържанието се създава и актуализира директно в базата данни на сървъра, обикновено собствениците не разполагат с резервно актуално копие на съдържанието, което пък води до рискове от загуба на информация в случай на срив в сървъра или хакерска атака на сайта.

\* Тъй като всеки динамичен сайт е базиран върху конкретна система за управление на съдържанието (CMS), изграждането на сайта трябва да бъде съобразено с възможностите и изискванията на тази система. Това обикновено води до ограничения както по отношение графичното оформление на сайта, така и по функционалностите му.

**Динамичен сайт е по-подходящ за вас ако:**

\* Планирате Вашият сайт да съдържа много десетки или стотици страници, които често (ежедневно или ежеседмично) ще бъдат променяни и обновявани.

\* Планирате изграждане на изцяло онлайн базиран бизнес (онлайн магазин).

\* Разполагате с необходимите ресурси (средства и персонал с необходимата подготовка) за редовна и компетентна поддръжка на сайта.

## 4. Бази данни

Базите данни са полезен и необходим инструмент при работа с всяка система, включваща манипулиране на голямо количество информация. В най-общия смисъл една база данни е компютърна система за регистриране и поддържане на логически свързани помежду си структури от данни.

**Видове бази данни**

Обикновено за потребителя една база данни представлява съвкупност от записи, съдържащи определена информация. Структурата на записа е свързана с елементи (атрибути, реквизити) на описанието на въвежданата информация. Елементите на описанието на записите от БД се определят от целите и задачите, поставени при събирането на конкретната информация. Понеже поставените цели и начините на събирането и обработването на информацията могат да бъдат различни, многообразието на съществуващите БД е много голямо, като в съвременната теория на БД са се утвърдили следните три модела:

А) йерархичен;

Б) мрежов;

В) релационен.

### 4.1 Йерархични БД

Дълго време те са били стандарт, като при тях се използват записи, свързани в дървовидна структура. Самото дърво е съставено от насочени надолу 1:N връзки. Потребителите трябва добре да познават конкретната структура, защото търсенето на данните в нея (запитването към БД) изисква команди, които да трасират дървото до необходимото ниво.

### 4.2 Мрежови БД

Мрежовата идея по същество е развитие на йерархичния метод за организиране на данни. При нея подмрежите от данни могат да се разглеждат като йерархично организирани данни. Мрежовата структура се описва с изходните и породените елементи от данни. Всеки елемент от данни може да се свързва с произволен друг елемент. Изискват сложно планиране и задълбочено познаване на естеството на данните, които се организират преди самото създаване на БД.

### 4.3 Релационни БД

При тях данните се представят като двумерни таблици, свързани помежду си с общи полета. Всяка таблица представя една релация. От логическа гледна точка релационните БД са множество от таблици, върху които могат да се извършват операции за търсене, извличане и обединяване на данните по колоните (полета) на таблиците. На потребителя е предоставена възможността да работи с данните, без да се интересува от физическото им реализиране.

### 4.4 Релационни БД и СУБД

Преобладаващата част от БД са релационни и следователно изградените за работа с тях СУБД (система за управление на бази от данни) също се основават на т. нар. релационен модел, изграден върху абстрактната теория за данните. Самите данни се манипулират чрез използване на операции от релационната алгебра които са:

Select – резултатът е нова релация, съставена от избрани редове от първичната релация;

Project - резултатът е нова релация, съставена от избрани колони на първичната релация;

Join - резултатът е нова релация, представляваща обединение на две първични релации върху обща колона.

СУБД осигуряват отличен начин за съхраняване и лесен достъп до сложна информация. Най-популярните търговски СУБД са Oracle, Sybase, Infornix, SQL Server на Microsoft и DB2 на IBM. Съществуват още няколко релационни бази от данни. Те принадлежат към класа софтуер познат като “софтуер с отворен код”. Към тях спадат MySQL, Postgres и продукта на Borland от 1999 година Interbase.

### 4.5. База данни MySQL

**MySQL** е проста, не усложнена база от данни за случаи, когато няма нужда от сложни търговски решения, а от добра производителност. Тя представя добри възможности за търсене в бази данни и за обслужване на Web-заявки, тъй като е бърза и надеждна. Освен това MySQL се доставя в реализации както за UNIX, така и за Windows.

Съществува обединение между РНР, MySQL и платформата Apache.

Основните части или слоеве на едно интернет-приложение за база данни са следните:

- Клиент - това е интернет-браузъра на потребителя, Java аплет, Java приложение или платформено зависима клиентска програма.

- Логическа част на приложението - кодирана в алгоритмите, използвани в CGI скриптове, специални модули на интернет-сървъра или зависим от приложението сървър.

- Свързване с базата данни - приложния програмен интерфейс на базата данни или основни протоколи за връзка с бази данни.

- Сървър за база данни.

Разработването на такива приложения може да бъде направено с използване на многослоен модел, тъй като един или повече от слоевете могат да бъдат обединени. Обикновено се използва трислойна система:

- Първи слой – уеб-клиент (например интернет-браузър).

- Втори слой – уеб-сървър, CGI-скриптове и приложни програмни интерфейси за връзка с базата данни (например Apache с модул РНР, поддържащ MySQL база данни и РНР скриптове).

- Трети слой – сървърът за база данни (например MySQL).

**Браузер**

**Java аплети/**

**приложение**

**Платформено**

**зависима**

**клиентска**

**програма**

**уеб-сървър**

**+**

**CGI**

**Програми**

**PHP скриптове**

ODBC/JDBC

Специфичен за базата

Данни API

**Бази данни**

**Втори слой**

***Трети слой***

*HTTP*

**HTTP**

**HTTP**

Фиг.*7*.Трислойна система на взаимодействие

## 5. Редактори и филтри за HTML

За потребители, за които е трудно да запомнят командите на езика HTML или искат да избегнат преправянето на вече създадени и форматирани с помощта на Word документи, са разработени специализирани редактори и филтри. Най-простият, но и най-ефективният начин за създаване на HTML страници е използването на текстов редактор за съставяне на кода и WWW клиент за неговата интерпретация. Този вариант изисква добро познаване на елементите на езика и е предпочитан от напредналите потребители.

Редакторите са програми, с които лесно могат да се създават HTML документи. Те се делят на текстови и графични. Много са подходящи редакторите, реализиращи режим WYSIWYG (What You See is What You Get). В такъв режим на работа авторът вижда непосредствено резултата от неговите действия. Някои редактори притежават собствена интерпретираща среда, други асоциират външни WWW клиенти за интерпретация на HTML кода.

Филтрите предоставят шаблони или макроси за различни, широко разпространени текстообработващи програми, които включват възможност за използване на командите на HTML или извършват преобразуване от вътрешния формат на текстообработващата система в HTML файл.

Използването на редактори и филтри може да улесни изучаването на езика HTML. Потребителят въвежда текст и разглежда непосредствено получения HTML код. Трябва да се отбележи, че редакторите и филтрите често не притежават най-новите възможности езика и на разпределената информационна система Web, което налага някои елементи да се въвеждат "ръчно” в съответния документ. Многократно нанесени корекции в един HTML документ, обикновено добавя или изоставя множество ненужни редове в кода на HTML страницата. Това е един страничен ефект при работа с редактиращите системи.

Изборът на редактор или филтър зависи от предпочитанията на потребителя. Всички те имат сходни възможности. За по-обикновени нужди за предпочитане са HTML филтрите. При необходимост от изграждане на по-сложен HTML проект трябва да се използват HTML редактори. Сложността зависи от броя на връзките, наличието на повече графични елементи, таблици, фреймови конструкции. Голяма част от HTML филтрите са freeware или shareware и са достъпни за копиране по Internet. Съществуват HTML редактори freeware или shareware, които също са общодостъпни. Редакторите с по-сериозни възможности обикновено са комерсиални.

## 6. CSS (Cascading Style Sheets )

За подобряване на форматирането на текстовата информация, изложена в приложението и за позициониране на обекти и създаване на рамки, са използват CSS (Cascading Style Sheets) - каскадни наборни стилове. Те позволяват да се прилага свойство или група от свойства към даден обект чрез прилагане на стил към този обект. Използвана е възможността за едновременно обновяване на всички обекти, съдържащи даден стил. Каскадните наборни стилове са част от динамичния HTML и за да може потребителя да ги вижда, е необходимо да притежава поне Internet Explorer 4.0. Въпреки че CSS не са част от HTML, те работят с HTML и предоставят възможност за по-голям контрол върху оформлението и представянето на web-страниците. С помощта на CSS, чиито стилови оформления се наричат още стилове, може да се указва на браузъра да третира всички тагове по един и същ начин, като така кодирането на страницата става по-просто и изчистено.Най-големият недостатък на CSS е липсата на пълна поддръжка от страна на браузърите. Докато Internet Explorer бързо се адаптира към повечето спецификации на CSS, Netscape не поддържа нито една от тези спецификации, освен в четвъртата си версия.

Възможностите на каскадните наборни стилове включват:

- Управление на печата - дават възможност за контрол относно начина, по който се отпечатва интернет-страницата;

- Допълнителен контрол над устройствата - дава възможност за предоставяне на инструкции за възпроизвеждането на страниците на устройства, различни от компютри и монитори;

- Динамично съдържание - дава възможност за промяна на съдържанието на страницата съобразно крайния потребител;

По-добро позициониране - предоставя инструменти, които могат да разположат елемент почти навсякъде в страницата.

## 7. PНР и HTML

Интернет все повече се използва за приложения повечето, от които включват работа с бази от данни. Тези сайтове и приложения са динамични, тъй като съдържанието им ще се променя в зависимост от данните, с които работят и действията на потребителя. В такива случаи се използва РНР. Чрез изпълняване на РНР програми на сървъра, може да се създават много мощни приложения, които си взаимодействат с базата данни и динамично генерират съдържание.

Основното различие между РНР и HTML страниците е в начина , по който работи с тях уеб-сървъра.

- Какво става с HTML страниците? - когато се получи заявка от браузъра за дадена страница, уеб-сървърът изпълнява три стъпки:

- прочита заявката на браузъра;

- намира исканата страница на сървъра;

- изпраща обратно тази страница чрез Интернет към браузъра

Това взаимодействие е представено на фиг. 4.



Клиент

**HTTP заявка**

**(**[**http://web**](http://web) **site)**

<HTML>

<B>This is web page</B>

</ HTML >

**Браузърът**

**интерпретира HTML кода и създава интернет-страницата**



**This is web page**

**уеб-сървър**

**HTTP**

**отговор**

**клиент**

Взаимодействие на HTML страниците със сървъра

- Какво става с РНР страниците? - вместо да се изпраща към потребителя статична HTML страница, искането е сървъра да извърши определени действия зависимост от потребителския РНР код: РНР ще вземе някои решения и ще създаде страница, която е подходяща за конкретна ситуация. Така, че когато се използва РНР, действията на сървъра са следните:

- прочита заявката от браузъра;

- намира исканата страница на сървъра;

- изпълнява инструкциите, заложени в РНР, за да модифицира страницата;

- изпраща страницата обратно чрез Интернет към браузъра.



Клиент

**HTTP заявка**

**(**[**http://web**](http://web) **site)**

*<HTML>*

*<? php PHP code ? >*

</ *HTML >*

**Браузърът**

**създава**

**страницата**



**Hello**

Web-сървър

**Получава страница**

Интерпретатор

на PHP кода

**<HTML>**

**<B> Hello </B>**

**</ HTML >**

Отговор на сървъра

**клиент**

Взаимодействие на PHP страниците със сървъра

Съществуват много неща, които могат да се направят с РНР и които не могат да се направят с HTML. Най-важната разлика е в това, че чистия HTML се интерпретира от браузъра, а не се изпълнява на сървъра. Примери:

- РНР улеснява редактирането на съдържанието на интернет-страница, като се актуализира съдържанието на базата данни, вместо да се променя самия HTML код;

- Създаване на страници, които ще бъдат настроени така, че да показват само нещата, от които се интересува конкретния потребител;

- Показване и модифициране на бази данни, съдържащи се в интернет-страницата и възможност за манипулации с данните в тях, като елементите се сортират в произволен ред или като се представя тяхно подмножество;

- Създаване на страници, които се ротират чрез поредица от различни графики;

Получаване на данни от потребител и връщане на информацията към потребителя в зависимост от тези данни.

### 7.1. PHP (PHP Hypertext Preprocessor)

PHP принадлежи на класа езици, познати като “мидълуер” (middleware). Тези езици работят плътно със сървъра за уеб, за да интерпретират запитванията, направени от World Wide Web, да обработят тези запитвания, да взаимодействат с други програми на сървъра, за да отговорят на заявките и след това да посочат на сървъра какво точно да предложи на браузера на клиента. Освен PHP съществуват няколко езика, които изпълняват подобна функция. Най-популярните са ASP, Perl и ColdFusion.

PHP представлява среда за писане на скриптове от страна на уеб сървъра, която се използва за създаване на динамични и интерактивни приложения. Всеки скрипт е последователност от команди (инструкции). За разлика от етикетите в HTML, които форматират текст или четат графични, видео или звукови файлове, командите в скриптовете предизвикват уеб сървъра да предприеме някакво действие.

За програмиране на PHP се използва обикновен текстов редактор (Notеpad на Windows, Emacs, vi или Kedit на Unix/Linux и др.), поради несъществуване на интегрирани среди за разработване (Integrated Development Environment – IDE) за PHP на този етап. Най-често файловете имат разширение .php (или .php4 при по-старите версии на PHP), но PHP може да интерпретира всичко, включително файлове .html.

Хубава черта на езика е, че той позволява движение между чистия HTML и команди, които са част от PHP. Работи се по следния начин: разделите в скрипта, намиращи се между отварящия етикет “<?php” и затварящия етикет “?>” ще се интерпретират от машината на PHP, а останалите части (извън тези етикети) ще бъдат третирани като обикновен код на HTML.

PHP страниците обикновено съдържат следните елементи: текст, HTML етикети и скриптове. Скриптовете се стартират когато браузърът ги изиска от уеб сървъра. На свой ред сървъра извиква PHP-интерпретатора, който чете извикващия го файл отгоре надолу и изпълнява всяка срещната команда от скрипта. Накрая препраща генерираната страница обратно към браузера.

С помощта на PHP могат да се решат следните задачи:

* създаване на интерактивни уеб страници;
* обработване на информация и данни за клиента, събрана от HTML форми;
* осъществяване на достъп и манипулиране с бази от данни и др.

PHP предлага добра възможност за бързо и лесно разработване на мощни приложения за уеб. Езикът има вградени множество от функции, които предоставят достъп до практически всичко. PHP е активен проект и целта на множеството от разработчиците му е подобряване на продукта почти ежедневно. Като софтуер на отворен код, той е безплатен и се разпространява чрез Internet.

### 7.2. Вградени библиотеки

PHP притежава голямо количество вградени функции за работа с:

голям брой база данни: Adabas, dBase, Informix, mSQL, MSSQL, MySQL, PostGreSQL, Oracle, Solid, Velocis и др.

Протокол LDAP (Light-weight Directory Access Protocol), който се използва за достъп до информация, свързана с директория;

XML (eXtensible Markup Language), отделящ съдържанието или информацията, която се пази в една Web страница, от нейното представяне;

Протоколи за електронна поща, като IMAP (Interactive Mail Access Protocol), използван за извличане на поща и SMTP (Simple Mail Transport Protocol), използван за изпращане на поща по Интернет;

Изображения;

PDF (Portable Document Format)

SNMP V3 (Simple Network Management Protocol Version 3), който се използва за постигане на контрол над мрежата.

Бисквитки и.т.н

### 

### 7.3. Конфигуриране на PHP

PHP може да бъде компилиран като:

* самостоятелен CGI интерпретатор – при това положение уеб сървърът създава инстанция на PHP интерпретатора, която влошава скрипта. Това обаче влошава представянето;
* като Apache модул – когато PHP е компилиран като Apache модул, той се изпълнява в същото адресно пространство като това на процеса на уеб-сървъра, което осигурява подобряване на представянето в сравнение с традиционните CGI интерпретатори, които са отделни процеси. Постоянна връзка с база данни не е налице, когато PHP се компилира като CGI интерпретатор, а само тук.

### 

### 7.4. Предимства на PHP

Висока производителност – PHP е много ефикасен. С него може да се обслужват милиони посещения на ден само с един евтин сървър. Резултатите от тестовете, публикувани от Zend Technologies (<http://zend.com>) показват, че PHP, надминава конкурентите си по производителност.

PHP е безплатен – PHP e софтуер с отворен код. Това означава, че няма ограничения в целите и предназначенията, за които се използва, както и за хората, които имат право да го използват. Потребителят може да прави и разпространява произволни модификации, заедно с оригиналния софтуер. В такъв случай потребителят може да вземе всяка част от кода, който е разработван от екипа по разработване на PHP, и да го използва като основа на неговия собствен продукт.

Междуплатформеност и преносимост – PHP е достъпен под много различни операционни системи. PHP работи на всеки популярен вариант на UNIX или Windows (обикновено без да се налага неговото модифициране) и е съвместим с трите водещи сървъра за Web – Apache, IIS (Internet Information Server), NES (Netscape Enterprise Server).

Лесен синтаксис – синтаксисът на PHP е базиран основно на програмните езици C и Perl, така че хора запознати със C, Perl или някой C-подобен език като Java и C++, ще могат почти веднага да пишат на PHP.

PHP е вграден в HTML – това означава, че страниците на PHP са обикновено страници, които „превключват” в режим PHP, когато е необходимо. Това от своя страна има много полезни последствия: PHP може бързо да бъде добавен към кода, създаден от някой редактор; PHP води до разпределяне работата между дизайнерите и авторите на скриптове; не е нужно всеки ред на HTML да бъде писан отново на език за програмиране; PHP може да намали цената на разработването и да увеличи ефективността; не е необходимо PHP да се компилира, той се интерпретира.

### 

### 7.5. PHP и HTML

Основното различие между PHP и HTML страниците е в начина, по който работи с тях уеб-сървъра:

При HTML страниците – когато се получи заявка от браузъра за дадена страница, уеб-сървърът прочита заявката на браузъра, след което намира исканата страница на сървъра и изпраща обратно тази страница чрез Интернет към браузъра.

При PHP страниците – вместо да се изпраща към потребителя статична HTML страница, искането е сървъра да извърши определени действия в зависимост от потребителския PHP код: PHP ще вземе някои решения и ще създаде страница, която е подходяща за конкретната ситуация. Така, че когато се използва PHP, сървърът прочита заявката от браузъра, намира исканата страница на сървъра, изпълнява конструкциите, заложени в PHP, за да модифицира страницата и изпраща страницата обратно към браузъра.

### 

### 7.5.1. Вграждане на PHP в HTML

Kодът на PHP е скрипт-код, вграден в HTML страницата, който се изпълнява на сървъра, преди да бъде изпратен към браузъра. По този начин PHP може да използва елементите на HTML. Всички програми, приложения и протоколи, които са съвместими с HTML от страна на клиента, са също така съвместими с PHP. Това означава, че може да се използва произволен метод за разработване на страници за интернет, а PHP да се прибави към тях. Съществуват два начина за добавяне на PHP код към HTML:

чрез използване на етикети – секциите на PHP в даден документ се означават чрез използването на специални етикети за PHP в началото и в края на всяка секция на PHP. Всичко, което се намира между тези етикети, се разбира от модула като PHP. Всичко извън тези етикети не засяга сървъра и ще бъде подминато и предадено на клиента за обработка. Съществуват следните етикети за PHP:

канонични - <?php тук кодът на php ?>

кратки отварящи - <? тук кодът на php ?>

етикети в стил ASP: <% тук кодът на php %>

етикети за скрипт на HTML - <script language=”php”> тук кодът на php </script>

чрез поставянето на PHP кода в отделен файл и извикването му.

## 8. World Wide Web

Информационната система World Wide Web (WWW или W3) е една от най-популярните услуги, достъпни чрез Internet. Тя позволява да се комбинират текст, аудио, видео, графика и анимация в мултимедийни документи. Хипервръзки в тези документи правят възможен достъпа до други документи, свързани с първичния. Те от своя страна могат да сочат към други документи, намиращи се в други сървъри по мрежата без да имат директна връзка помежду си.

Информационната система World Wide Web е базирана на хипертекст (hypertext) технология. Една дума в хипертекст документ може да служи като указател (hyperlink) към друг документ, в който се намира информация свързана с думата-указател. В съвременните Web документи свободно се комбинират хипервръзки, аудио и видео фрагменти, графични икони и изображения, което ги превръща в хипермедийни.

Информационната система WWW се състои от множество информационни сървъри (Web server), които са постоянно достъпни по мрежата и непрекъснато се променят. Наблюдаваните промени са насочени към съдържанието на представената информация, начина на нейното представяне и структуриране. Нови Web точки непрекъснато се появяват в Internet.

За получаването на достъп до WWW е необходима връзка с мрежата и програма клиент (client, browser), която интерпретира и визуализира документите, достъпни чрез WWW. Документите са хипермедийни и съдържат текст и команди за структуриране. По този начин WWW клиентът извършва форматиране с цел получаване на най-добрите възможни визуални резултати върху екрана на компютъра.

Протоколът HTTP

Една от целите на проекта WWW е лесното получаване на информация, независимо от това къде се намира по Internet мрежата. Като стандартен формат за представяне на WWW документи се използва хипертекст. Протоколът HTTP (HyperТext Transport Protocol) е прост комуникационен протокол за извличане на документи на базата на TCP/IP протоколи от Web сървъри по Internet. Той се базира на факта, че документите, които се извличат, съдържат информация за бъдещите връзки, които потребителят може евентуално да заяви (за разлика от FTP или Gopher, при които информацията за възможните бъдещи връзки трябва да бъде предадена чрез протокола).

Протоколът за пренасяне на хипертекст HTTP притежава необходимата простота и скорост за разпределена кооперативна хипермедийна информационна система (Intranet). В такава система е необходимо бързо проследяване на връзки между информационни единици, които се намират върху отдалечени станции. Времето за отговор трябва да бъде от порядъка на 100 ms за проследена връзка, което налага изискването за бързина при извличането на информация. Трябва да се вземе под внимание, че информационните системи, използвани в практиката, изискват повече функции, включващи търсене, актуализация и анотация.

Протоколът HTTP е обектно-ориентиран и може да се използва в разпределени обектно-ориентирани системи. Важно негово свойство е универсалност на представяне на данните, което позволява системите да са независими от нови, по-усъвършенствани методи за представяне на данни. Протоколът HTTP позволява използването на отворено множество от методи. Той се базира на достъпа, осъществяван от Universal Resource Identifier URI към който се включват - име - URN - Uniform Resource Names (виж RFC 1737) или адрес - URL - Uniform Resource Locators (виж RFC 1738) (виж Забележка).

Компонентите на протокола HTTP са:

\* изпращане на заявка до сървъра;

\* метод, приложен върху обекта, специфициран чрез URL;

\* заглавна информация (list of headers) в съобщението-заявка;

\* получаване на отговор;

\* кодове на състоянието;

\* заглавна информация (list of headers) за всеки предаван обект;

\* съдържание на всеки предаван обект;

\* формати.

За работа с WWW не са необходими знания върху HTTP. Протоколът се обслужва от съответните приложения, както от страна на сървърите, така и от страна на клиентите. Трябва да се отбележи, че този протокол обуславя поддръжката на информационни услуги в Internet и не може да се разглежда като конкурент на останалите видове основни протоколи, поддържащи съответните услуги. Именно това съображение превръща HTTP протокола в платформа при комуникацията между потребители/услуги и proxies/gateways и други Internet протоколи като SMTP, NNTP, FTP, Gopher, WAIS с което осигурява хипермедиен достъп до ресурси на различни приложения и с това улеснява интегрирането на услугите. Протоколът Hypertext Transfer Protocol - HTTP/1.0 е подробно описан в RFC 1945 (виж Забележка).

Забележка: Съкращението RFC (Request for Comments) представлява рубрика, поддържана от специализирана работна група (Network Working Group), с описание на всички технически аспекти на Internet протоколите.

Инсталиране на WWW сървър

Един WWW сървър (Web server) се създава на базата на инсталиран HTTPD (daemon) програмен пакет, реализиращ HTTP протокол. Функцията на сървъра е да обработва и изпълнява получените заявки за достъп до съдържащата се в неговата структура информация от WWW клиенти. Достъпът до хипермедийни документи в Internet се осъществява чрез протокол HTTP с помощта на HTTPD сървър (WWW сървър). Съдържанието на извлечения от един WWW сървър документ посредством WWW клиент се визуализира на екрана на локалния компютър на потребителя. Не е възможно в тази схема (клиент-сървър) два WWW клиента да обменят документи директно помежду си.

Първата стъпка при инсталиране на WWW сървър е да се определи хардуерната платформа и операционната система, които ще се използват. Програмно осигуряване за WWW сървъри е достъпно за всички платформи (персонални компютри и техните разновидности, работни станции), така че изборът се базира на съображения за икономия, удобство и съвместимост със съществуващите машини в локалната мрежа. Важно е да се отбележи, че за WWW сървър не е необходимо да се избира твърде мощен компютър. Главното предназначение на сървъра е просто да съхранява и изпраща файлове. Ето защо скоростта на мрежата е много по-важна отколкото мощността на процесора, което трябва да се свързва и с избор на подходящи мрежови контролери. Друго изискване е по отношение обема на дисковата памет. Ако потребителят иска дългосрочно решение (2 години) добре би било да се осигури външна дискова памет не по-малка от 2.5 TB. Обемът оперативна памет е фактор, който често се пренебрегва за сметка на бърз процесор. Ефективната производителност на един WWW сървър е в пряка зависимост от обема RAM и няма отношение към дългосрочността на поставените цели или изпълняваните проекти. Оперативна памет над 32 GB е задължителна, обем памет над 64 GB е препоръчителна.

Втората стъпка е да се осигури връзка на избрания компютър с Internet. Компютърът трябва да се включи към мрежата чрез интерфейсна мрежова карта (платка) и съответните програмни драйвери. В допълнение към това е необходимо в компютъра да се инсталира софтуерният стек на протокола TCP/IP. Софтуерът за TCP/IP трябва да се конфигурира с IP адреса на компютъра и IP адресите на машините - Gateway (шлюз) и DNS сървъри (сървър за разрешаване на съответствията между IP адрес и символично име на машина). Необходимо е също да се актуализират таблиците на всички рутери в мрежата, така че TCP/IP трафикът от новосъздадения сървър да достига Internet. Това включва също установяване на адреса на подходящия за мрежата proxy сървър. Сървърите DNS (Domain Name Server) за съответния домейн (област) трябва да се актуализират с името, идентификатора и IP адреса на компютъра, който става сървър.

Инсталирането на софтуера за TCP/IP е твърде различно за различните платформи.

При UNIX станциите поддържането на TCP/IP е част от дефиницията на работната станция. Конфигурацията по подразбиране обикновено работи добре.

Microsoft Windows NT и Windows 95 включват TCP/IP стек, но програмата Setup не го инсталира автоматично. Ядрото на TCP/IP се конфигурира като самостоятелна стъпка, след като системата вече е пусната в действие. Конфигурирането става от управляващия панел (Control Panel|Network) на Windows 95/NT. В процеса на конфигуриране може да се използва помощна информация (on-line help).

Сравнително по-труден е този процес за Macintosh системи (използува се TCP/IP стек на Apple - MacTCP).

Потребителите на Microsoft Windows 3.11 трябва да прибягнат до допълнителен пакет - TCPIP32 или да закупят някой от TCP/IP пакетите за Windows.

За потребители непрограмисти и такива, които нямат опит в работа с мрежи, най-добре е свързването към нея да се прави съвместно с персонала, администриращ мрежата или с организацията, осигуряваща Internet услуги.

Третата стъпка е да се осигури програмен пакет за инсталиране на WWW сървър, предназначен за конкретния компютър и операционна система. Това може да стане най-лесно чрез самата работна група W3 - адресът на началната страница на CERN е:

http://info.cern.ch/hypertext/WWW/TheProject.html

където, може да се избере необходимата за конкретния случай страница с информация за различни програмни среди и програми. Достъпът посредством FTP протокол се осъществява на следния адрес:

ftp://ftp.w3.org

Софтуерът и документацията са достъпни обикновено във вид на архивиран файл с формат .Z за UNIX системи, .SIT за Macintosh системи, .ZIP за Windows 3.x и Windows 95/NT. Например файлът, предназначен за Windows 3.x е SERWEB03.ZIP.

За Windows NT 4.0 Server, WWW сървърът (Microsoft Internet Information Server) е част от дистрибутивния пакет на системата и неговата инсталация е подробно документирана. Допълнителна информация се намира на следния адрес:

http://www.microsoft.com/infoserv

За Windows 95 може да се ползва OmniHTTPd 1.0. Продуктът е freeware и може да се изтегли от следния адрес:

http://www.fas.harvard.edu/~glau/httpd/

Четвъртата стъпка е дезархивиране на файла и пускане на сървъра. Дезархивирането става с помощта на специфична за съответната система програма. Процедурата за инсталиране и създаване на начална страница на новия WWW сървър е описана в съпровождащата документация, както и една примерна начална страница, с която сървърът да започне работа.

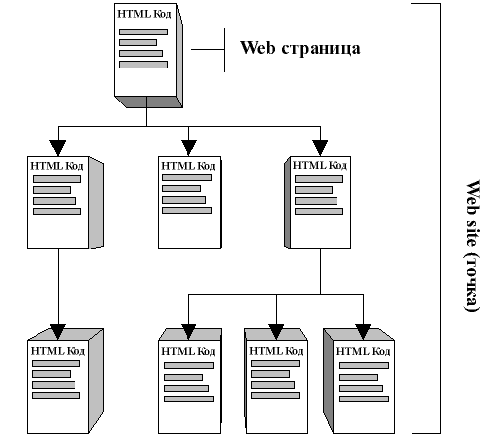
В Internet може да се намери богата информация със сравнителни таблици за характеристиките и приложимостта на различни видове програмни пакети за Web сървъри. На адреса на Web сървъра Webcompare се намира богата информация за множество продукти - freeware, shareware или комерсиални версии. Този Web site е достъпен на следния адрес:

http://webcompare.iworld.com

В популярния Web сървър Yahoo също може да се намери подробна информация за различни видове WWW сървъри и тяхното приложение за различни платформи на следния адрес:

http://www.yahoo.com/Computers\_and\_Internet/Internet/World\_Wide\_Web/HTTP/Servers/

Фигура 1 - Структура на Web сървър



Изобразяването на Web страници и тяхното структуриране и създаване са различни неща. Един WWW клиент интерпретира и изобразява Web страницата по начина, по който Web сървъра я изпраща. В сървъра се съдържа цял Web проект, реализиран на базата на множество Web страници, написани на езика HTML. Връзките между тях се задават във всяка изобразавана Web страница под формата на хипервръзки (hyperlinks). Следователно един Web сървър съдържа един HTML проект, обикновено наричан Web site (Web точка). Изобразеното на Графика 1 дава нагледна представа за структурата на една Web точка.

Глава 3

Описание и реализация на проекта

„Интернет портал за технологични новини”

## Предварително планиране

Преди започването на работата по даден уеб сайт, всеки уеб разработчик трябва да си отговори на няколко кардинални въпроса, които ще имат значение по-нататък в разработката на проекта.

1. За кои уеб браузъри да се осигури поддръжка?

Понастоящем най-разпространените уеб браузъри в света са:

* Internet Explorer (Trident rendering engine);
* Mozilla Firefox (Gecko rendering engine);
* Google Chrome (WebKit rendering engine);
* Opera (Presto! Rendering engine);
* Apple Safari (WebKit rendering engine)

Организацията, която е създала и поддържа стандартите в World Wide Web се нарича W3 Consortium (W3C) и е създадена от Тим Бърнърс-Лии, човекът, който в началото на 90-те пусна първият уеб сървър в CERN. Стандартите, които развива W3С, имат за цел да унифицират и улеснят разработката на уеб сайтове, и поддръжката им от различни уеб браузъри. В исторически план не всички компании, разработващи уеб браузъри, са приемали с охота стандартите на W3C. Типичен пример за това е Майкрософт и техният Internet Explorer. До версия 6 IE почти не следваше вече общоприетите по това време уеб стандарти на W3C, а следваше измислени „правила” (стандарти е силно казано), създадени от Майкрософт, като това силно затрудняваше уеб разработчиците. От версия 7 нагоре, Internet Explorer започна все повече да следва стандартите на W3C под натиска на общността от уеб разработчици. Настоящата версия на Internet Explorer – 9 за щастие поддържа повечето W3C стандарти

Като се има предвид, че в днешно време и петте „големи” браузъра поддържат повечето W3C стандарти, въпросът за избор на това кои браузъри да поддържа сайта изглежда малко поостарял. Минаха времената, когато най-долу в уеб сайтовете пишеше „Looks best in IE, 800x600 or higher.” От тук идва следващият важен въпрос:

1. Модерен изглед или поддръжка на стари браузъри (backward compatibility) ?

За да си отговорим на този въпрос трябва да помислим за каква аудитория е предназначен сайта ни. Дали е за корпоративна аудитория, където компютрите най-често са неглижирани и са с вградения в Windows Internet Explorer? Същото важи и ако таргетираме аудитория от възрастни хора, или хора с ниски ИТ познания. В този случай неминуемо ще трябва да лишим сайта си от определена доза интерактивност и модерен изглед и да се съсредоточим върху това какво поддържат IE 6, 7 или 8.

Ако пък аудиторията ни е от компютърни ентусиасти, при които винаги всичко е “up-to-date” няма да се грижим толкова за съвместимостта назад, а ще наблегнем на богатото съдържание и модерния изглед. Като се има предвид, че един уеб портал, съдържащ новини от сферата на ИТ ще е таргетиран именно към общността от ентусиасти, аз реших да осигуря поддръжка на най-новите версии на петте най-разпространени браузъра:

* Internet Explorer 9+
* Mozilla Firefox 3.6+
* Google Chrome 10+
* Opera
* Apple safari

## Разработка на уеб сайта – стъпка по стъпка

Уеб сайтът се поддържа от Wordpress CMS (content management system) и се състои от следните по-важни файлове:

* header.php – генерира хедър частта. Тя е еднаква за всички страници. Съдържа двете главни менюта и логото.
* footer.php – генерира футъра. Отново е еднакъв за всички страници.
* index.php - генерира началната (главната) страница;
* single.php – генерира страница с единична статия;
* page.php – генерира единична вътрешна страница, която не е статия. Например Contacts, About us и др.
* sidebar.php – генерира сайдбара.
* comments.php – съдържа PHP програмна логика за работа с коментари към статиите;
* functions.php – съдържа PHP програмна логика, чрез която моят сайт декларира пред Wordpress какви функции поддържа;
* 404.php – генерира страницата, която се визуализира, когато се получи заявка за ресусрс, който не съществува. Изписва съобщение за грешка “Error 404 – Not Found”
* База данни MySQL – пази всички статии, категории, коментари, тагове и др. Управлява се чрез PHPMyAdmin
* style.css – главният файл с CSS стиловете.

## Структура на сайта

В най-горната част от сайта се намира меню-навигация (Top menu) към вътрешните страници на сайта – Contacts, About us, Friend sites, Contribute и Terms and Conditions. Реализирал съм го като падащо меню, при което основните меню елементи са линкове към самите страници, а подменютата са линкове към определени секции са тези страници. Например меню елементът About съдържа линкове към подсекциите History и Mission на страницата About.



В HTML менюто е реализирано на следния принцип:

<nav>

<ul id=”top-menu”>

<li>

<a href=”link-to-internal-page”>

<ul id=”sub-menu>

<li>…</li>…<li></li>

</ul>

<li></li>

…

</ul>

<nav>

Използвал съм CSS3 елементът <nav>. Той служи за семантично разделение на навигационните менюта от останалата част на сайта. Също така, ако сайта се чете през screen reader (устройство за слепи хора), частта с менюто се пропуска.

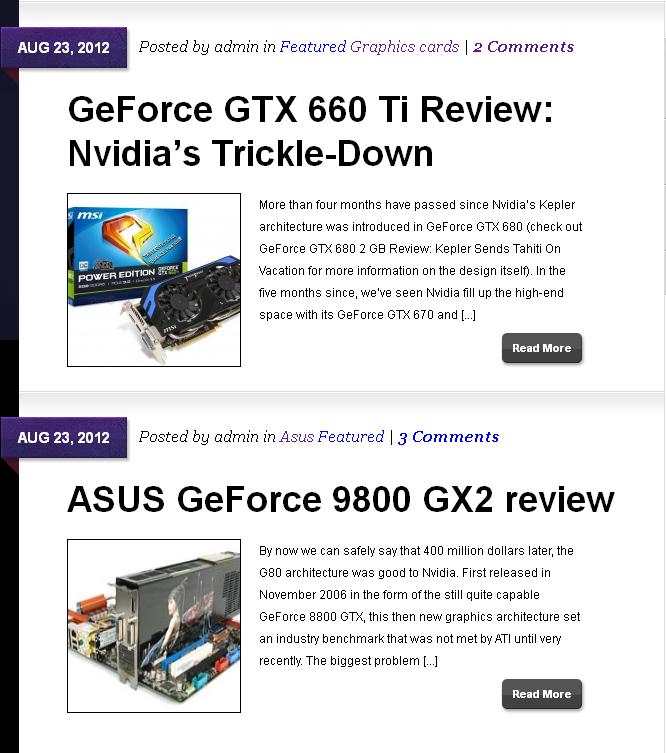
По-надолу е логото, а под него е главното меню с категориите. (Main menu) To e реализирано също както Top menu, само че съдържа линкове към категории. Съдържанието на сайта е разделено на категории според типа на продукта, за който е статията/ревюто. Основните категории са пет:

* Processors;
* Motherboards;
* Graphic cards;
* Storage;
* Desktop cases
* Featured

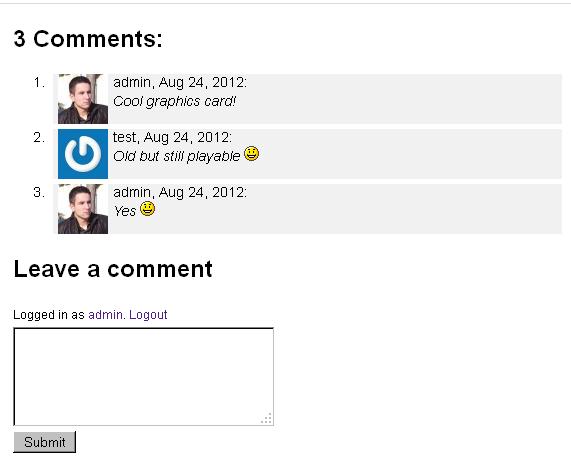
Всяка от тези категории е разделена на подкатегории в зависимост от фирмата-производител на съответния продукт. Всяка статия може да бъде в една или повече категории, примерно статията „Antec P280 Super Case Review” може да бъде едновременно в Desktop cases и във Featured. Първите три статии в категория Featured се визуализират в т. нар. “Featured Articles Panel” в горната част на сайта.

Освен категориите като начин за таксономия използвам и тагове. Таговете като принцип са въведени с Web 2.0. При писането на една статия авторът може да посочи един или повече тага (ключови думи), които са всъщност най-важните думи в статията и дават представа за съдържанието й. Примерно в статията „ASUS GeForce 9800 GX2 review” подходящи тагове са asus, geforce, videocard, gpu, 9800, gx2, “graphics card”, nvidia. Изписват се с малки букви и са разделени със запетаи.

Началната страница на сайта (генерирана чрез index.php) съдържа откъси от статии, придружени от заглавията си и заглавна картинка (thumbnail).

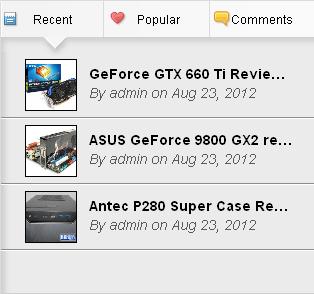


При кликване върху заглавието или картинката, потребителят бива отвеждан до страницата на съответната статия, генерирана от файла single.php. В тази страница е пълният текст, заглавната картинка, както и други картинки, свързани със статията. Картинките са в умален размер, като при кликване върху всяка една от тях на потребителя се предоставя съответната картинка в пълен размер. В горната част се вижда датата на публикуване, автора и категорията. В долната част са таговете и блокът с коментарите. За да може потребителят да използва аватар при писаните от него коментари, трябва да има регистрация в [www.gravatar.com](http://www.gravatar.com) и да качи желана от него снимка за аватар. При оставянето на коментар от потребителя се изисква да въведе e-mail, като ако той съвпада с e-mail-а, регистриран в [www.gravatar.com](http://www.gravatar.com) се визуализира съответният аватар.

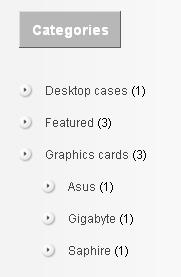


В дясната част на сайта е сайдбара (sidebar), той е еднакъв за всички вътрешни страници. На него има няколко инструмента (уиджети). Те могат да бъдат променяни от контролния панел на сайта, като ще разгледаме това по-надолу.

Първият уиджет предоставя линкове към най-новите статии (Recent), към най-коментираните статии (Popular) и към последните коментари (Comments).



По-надолу е списъкът с категориите, които са със съответното йерархично отместване надясно. До всяка категория пише колко статии има в нея. При кликване върху дадена категория се визуализират всички статии, които са в тази категория.



Най-долу в сайдбара е тъй натеченият tag cloud. Това е списък с всички тагове, като колкото по-голям е шрифта на даден таг, толкова по-често се среща той. При кликване върху даден таг се визуализират всички статии, които го съдържат.



## 

## Управление на съдържанието на сайта

В Wordpress има няколко типа потребители, като най-важните са администратор и автор. Администраторът има пълната власт да прави всичко със сайта. Има достъп до настройки, писане на статии, триене, контрол на потребителите и др. Обикновено е програмист.

Авторът може само да пише статии, да ги редактира и да ги трие. Обикновено е клиент, който няма високи ИТ познания.

Всичко в сайта подлежи на модифициране и промяна от страна на администратора или автора, като това се извършва от контролния панел на сайта, достъпен на адрес [www.bakalov.cloudvps.bg/wp-admin](http://www.bakalov.cloudvps.bg/wp-admin).

Контролът на потребителите се осъществява от меню Users.

Писане на нова статия

Избираме Posts 🡪 Add new. Тук имаме текстово поле с много опции. Може да избираме между нормален изглед в стил MS Word и HTML изглед. Можем да вмъкваме картинки, видео клипчета и др. За да добавим заглавно изображение към статията, кликваме долу в дясно на Featured image. Това изображение се използва като thumbnail навсякъде, където има препратка към съответната статия. Също е добре да добавим тагове и да посочим в коя категория да бъде статията. Всичко това става от десния панел. Когато сме готови натискаме бутона Publish.

Редактиране на статия

От менюто Posts избираме All posts и след това кликваме върху статията, която искаме да редактираме. Когато сме готови с редактирането натискаме бутона Update.

Изтриване на статия

От менюто Posts 🡪 All Posts посочваме с мишката статията, която изкаме да изтрием и натискаме Trash. Така тя отива в кошчето. Ако искаме да я изтрием окончателно, избираме Trash от горната част на страницата със статии и я изтриваме и от там.

Управление на главните менюта

Съдържанието на описаните по-горе Top menu и Main menu се контролира от Appearance 🡪 Menus. В горната част на екрана потребителят избира менюто, по което иска да прави промени – Top menu или Main menu. Потребителят може да добавя към менюто линкове към определени статии, към категории, към вътрешни страници или към външни уеб сайтове. В ляво са възможните меню елементи, които могат да бъдат добавяни, а вдясно е йерархичната им подредба в менюто. За да се направи даден меню елемент да е „дете” на друг, е необходимо да се отмести леко надясно. Това автоматически създава нов <ul> елемент в родителското <li> и поставя в него съответните елементи-деца под формата на <li>-та.

Управление на страничната лента (сайдбар)

Както вече споменахме, сайдбарът съдържа елементи, наречени уиджети. Потребителят може да контролира какви и колко уиджета да има на страничната лента. Tова става от меню Appearance 🡪 Widgets.

В лявата страна на екрана са всички възможни уиджети, които могат да бъдат добавени. Когато потребителят иска да добави уиджет към сайдбара, трябва да го довлачи до дясната част на екрана (Default right sidebar). При пускането му там, уиджета се отваря и на потребителя се предоставят различни опции за настройка.

Има един по-специален уиджет, който съм включил по подразбиране в сайдбара. Това е т.нар. Tabbed Widget (уиджет с раздели). Разделите на този уиджет всъщност са други уиджети. Можем да добавим всеки уиджет като раздел на Tabbed Widget като го пуснем в полето „Invisible Widget Area”. След това от настройките на Tabbed Widget избираме в кой раздел да бъде съответният „подуиджет”, заглавието му и някои други опции.

Tabbed widget поддържа един, два или три раздела. Всеки от тези раздели може да има неограничен брой елементи (list items), като ако са прекалено много вдясно се появява скролбар.

Бекъп на цялото съдържание и базата данни

Това става от менюто Tools 🡪 Export. Създава се XML файл, който съдържа цялата информация от уеб сайта. Този файл след това може да се импортира от менюто Tools 🡪 Import

## Защита на дизайна от счупване при произволно съдържание.

Всеки уеб разработчик трябва да прави сайтовете си по такъв начин, че да не се появяват дефекти в дизайна (счупване) при пълненето на сайта с реално съдържание. Не трябва да се допуска на дадени места текста да излиза от рамките си, прекалено дългите думи трябва да се пренасят адекватно и прекалено многото меню елементи не трябва да чупят менюто.

На заглавията на статиите в CSS съм задал overflow: hidden и max-height толкова, колкото да са най-много на два реда. Ако са по-дълги остатъка им се скрива. На откъсите от статиите също съм им задал тези две CSS пропъртита. Когато потребител иска да прочете цялата статия, трябва да кликне върху заглавието и така отива до съответната вътрешна страница със статията.

На датата, и полето с автора и категорията съм задал

overflow: hidden;

white-space: nowrap;

text-overflow: ellipsis;

По този начин ако текста не може да се вмести в контейнера си, се реже и се появява многоточие. Моля, имайте предвид, че горните три пропъртита, за да визуализират многоточие, трябва винаги да се използват и трите, и текста да е на един ред.

Горните три пропъртита се използват практически навсякъде в сайта, където имаме един ред текст и искаме ако той е прекалено дълъг да се проявява многоточите. Използвал съм ги в сайдбара, за разделите и съдържанието на Tabbed Widget, заглавията на уиджетите Categories и Tags, както и за списъка с категориите в уиджета Categories.

Всички изображения в сайта имат зададена ширина и височина както в HTML-a, така и в CSS-a. Имат и alt и title атрибути, и са оптимизирани за добър компромис между качество и размер.

Заключение

Създаденият в настоящата дипломна работа уеб сайт следва най-новите принципи и стандарти в уеб дизайна. В момента е запълнен със съдържание на IT тематика, но лесно може да се използва за произволно съдържание, от всички сфери на общественият живот. Постарал съм се да осигуря поддръжка за всички модерни уеб браузъри, като в същото време представям съдържанието по максимално богат и разчупен начин. С добавянето на ново съдържание може да се справи всеки, дори и да няма компютърни познания – управлението чрез администраторския панел е достатъчно интуитивно и лесно.

Съдържанието е добре организирано в раздели и категории и потребителят лесно може да се ориентира и да намери каквото му трябва

За в бъдеще уеб сайта може да се оптимизира за работа с мобилни телефони и таблети. За тази цел трябва да се внедри тъй-нареченият responsive design. Този тип дизайн разчита най-вече на Media Queries – това са CSS3 техники, които, на базата на резолюцията на конкретния дисплей, изобразяват съдържанието по-различен начин, и така дадено съдържание може да се направи да изглежда различно на устройства с различни резолюции, улеснявайки употребата на сайта.

Пожелавам ви приятно сърфиране в моя уеб сайт!

Приложение 1 – сорс кодове

## header.php – файлът, който генерира хедър частта на сайта

<!DOCTYPE html>

<html <?php language\_attributes(); ?>>

<head>

<meta charset="UTF-8" />

<meta name="description" content="Vladimir Bakalov PC News" />

<meta name="keywords" content="vladimir,bakalov,vladimir bakalov,web design,webdesign,web-design,project, pc news," />

<link rel="stylesheet" type="text/css" href="<?php bloginfo('stylesheet\_url'); ?>" />

<title>

<?php

if (function\_exists('is\_tag') && is\_tag()) {

single\_tag\_title("Tag Archive for &quot;"); echo '&quot; - '; }

elseif (is\_archive()) {

wp\_title(''); echo ' Archive - '; }

elseif (is\_search()) {

echo 'Search for &quot;'.wp\_specialchars($s).'&quot; - '; }

elseif (!(is\_404()) && (is\_single()) || (is\_page())) {

wp\_title(''); echo ' - '; }

elseif (is\_404()) {

echo 'Not Found - '; }

if (is\_home()) {

bloginfo('name'); echo ' - '; bloginfo('description'); }

else {

bloginfo('name'); }

if ($paged>1) {

echo ' - page '. $paged; }

?>

</title>

</head>

<body>

<div id="site-wrap">

<!-- HEADER -->

<div id="header">

<!-- The top drop-down menu -->

<nav>

<?php wp\_nav\_menu( array( 'sort\_column' => 'menu\_order', 'menu\_id' => 'top-menu', 'theme\_location' => 'top-menu', 'container' => false ) ); ?>

</nav>

<!-- Site Logo -->

<a href="<?php echo get\_option('home');?>" id="logo" title="Go to Home">

<img src="<?php echo get\_template\_directory\_uri(); ?>/images/logo.png" alt="Site Logo" width="156" height="46" />

</a>

<!-- The main drop-down menu -->

<nav>

<?php wp\_nav\_menu( array( 'sort\_column' => 'menu\_order', 'menu\_id' => 'main-menu', 'theme\_location' => 'main-menu', 'container' => false ) ); ?>

</nav>

</div> <!-- closing header -->

<!-- End of Header -->

## index.php – главният файл на уеб сайта

<?php get\_header(); ?>

<!-- INDEX.PHP -->

<!-- MAIN CONTENT -->

<div id="main-content">

<div id="jagged-border-top"></div> <!-- In this div I use sprites, check the CSS -->

<div id="background" class="clearfix">

<!-- The featured articles panel -->

<div id="featured-wrap" class="clearfix">

<ul id="featured">

<?php

/\* Because I use 2 loops on this page,

for the first loop I'll use a variable ($my\_query),

because I don't want to disrupt the second loop \*/

?>

<?php $my\_query = new WP\_Query('category\_name=featured&posts\_per\_page=3'); ?>

<?php if($my\_query->have\_posts()): ?>

<?php $count=1;?>

<?php while ($my\_query->have\_posts()) : $my\_query->the\_post(); ?>

<li>

<a href="<?php the\_permalink();?>" id="link\_<?php echo $count; ?>" title="<?php the\_title();?>">

<?php the\_post\_thumbnail(array(79, 79)); ?>

<div class="text-holder">

<h1><?php the\_title(); ?></h1>

<span class="author-featured">Posted by <?php the\_author(); ?> on <?php the\_time(get\_option('date\_format')); ?></span>

</div>

<div class="featured-nav-arrow"></div>

<div id="preview\_<?php echo $count; ?>" class="preview">

<div class="begining"></div>

<div class="middle">

<p><span>&ldquo;</span><?php echo get\_the\_excerpt(); ?>.&ldquo;</p>

</div>

<div class="ending"></div>

</div>

</a>

</li>

<?php $count++; ?>

<?php endwhile; ?>

<?php endif; ?>

</ul>

</div>

<!-- ARTICLES -->

<div id="faux-columns" class="clearfix">

<!-- The "Faux columns" technique. I used it to make the sidebar look as tall as the main content -->

<div id="article-holder" class="clearfix">

<?php if(have\_posts()): ?>

<?php $post\_count=1; //The first 2 posts are big, and the following 4 are small. This variable controls how posts are being displayed. ?>

<?php while(have\_posts()): ?>

<?php if ($post\_count <= 2): ?>

<?php the\_post();?>

<div class="big-article clearfix">

<div class="big-date">

<?php the\_time(get\_option('date\_format')); ?></div>

<div class="big-article-author">

Posted by <?php the\_author();?> in <?php the\_category(' ');?> | <a href="<?php the\_permalink();?>#comments-wrap" class="comments-count" title="Go to comments"><?php comments\_number();?></a>

</div>

<a href="<?php the\_permalink(); ?>" class="article-heading" title="Go to post">

<h1><?php the\_title(); ?></h1>

</a>

<a href="<?php the\_permalink(); ?>" title="Go to post">

<?php the\_post\_thumbnail(array(172, 172)); ?>

</a>

<?php the\_excerpt(); ?>

<a href="<?php the\_permalink(); ?>" class="read-more">

<span class="read-more-beginning"></span>

<span class="read-more-content">Read More</span>

<span class="read-more-end"></span>

</a>

</div>

<?php $post\_count++; ?>

<?php else : ?>

<?php the\_post(); ?>

<div class="small-article clearfix">

<div class="small-date">

<?php the\_time(get\_option('date\_format')); ?>

</div>

<div class="small-article-author">

Posted by <?php the\_author();?> | <a href="<?php the\_permalink();?>#comments-wrap" class="comments-count" title="Go to comments"><?php comments\_number(); ?></a>

</div>

<a href="<?php the\_permalink(); ?>" class="article-heading" title="Go to post">

<h1><?php the\_title(); ?></h1>

</a>

<a href="<?php the\_permalink(); ?>" title="Go to post">

<?php the\_post\_thumbnail(array(82, 82)); ?>

</a>

<?php the\_excerpt(); ?>

</div>

<?php endif; ?>

<?php endwhile; ?>

<?php wp\_pagenavi(); ?>

<?php else: echo "No Posts"; // This is in case there are no posts. ?>

<?php endif; ?>

</div> <!-- closing article-holder -->

<?php get\_sidebar(); ?>

<?php get\_footer(); ?>

## sidebar.php – генерира динамичната странична лента

<!-- SIDEBAR -->

<div id="sidebar" class="clearfix">

<?php if (function\_exists('dynamic\_sidebar') && dynamic\_sidebar('default\_right\_sidebar\_1')) : ?>

<!-- The dynamic sidebar's contents will be automatically generated here -->

<?php else : ?>

<!-- Put here the static stuff. It will be displayed if you don't have a dynamic sidebar -->

<?php endif; ?>

</div> <!-- closing sidebar -->

</div> <!-- closing faux-columns -->

</div> <!-- closing background -->

<div id="jagged-border-bottom"></div> <!-- In this div I use sprites, check the CSS -->

</div> <!-- closing main-content -->

## footer.php – генерира футъра и затваря по-горе отворени div елементи

<!-- FOOTER -->

<div id="footer">

Copyright &copy; 2012 Vladimir Bakalov. All rights reserved.

</div> <!-- closing footer -->

</div> <!-- closing site-wrap -->

<?php wp\_footer();?>

</body>

</html>

Използвана литература

[www.php.net](http://www.php.net)

[www.phpacademy.org](http://www.phpacademy.org)

[www.phptherightway.com](http://www.phptherightway.com)

[www.codex.wordpress.org](http://www.codex.wordpress.org)

[www.lynda.com](http://www.lynda.com)

Лекции и учебни материали по Web front-end development на фирма

Телерик – София: <http://academy.telerik.com/student-courses/web-design-html-5-css-3-javascript/resources>

[www.wikipedia.org](http://www.wikipedia.org)

Съдържание

Увод……………………………………………………………………..………………….2

Глава 1 Основни характеристики на уеб приложенията………………….……….3

[История на Интернет 3](#_Toc334205555)

[Отличителни черти на световната компютърна мрежа Интернет 5](#_Toc334205556)

[Термини при работа в Интернет 6](#_Toc334205557)

ГЛАВА 2 Същност, техники и инструменти за разработване на Интернет приложения…………………………………………………………………………………10

[Уеб-сървъри 11](#_Toc334205558)

[Страници и уеб-браузъри 12](#_Toc334205559)

[Статични и динамични интернет-сайтове 12](#_Toc334205560)

[Статични интернет-сайтове 12](#_Toc334205561)

[Динамични технологии 14](#_Toc334205562)

[Бази данни 16](#_Toc334205563)

[Йерархични БД 17](#_Toc334205564)

[Мрежови БД 17](#_Toc334205565)

[Релационни БД 17](#_Toc334205566)

[Релационни БД и СУБД 18](#_Toc334205567)

[База данни MySQL 18](#_Toc334205568)

[Редактори и филтри за HTML 20](#_Toc334205569)

[CSS (Cascading Style Sheets ) 21](#_Toc334205570)

[PНР и HTML 22](#_Toc334205571)

[PHP (PHP Hypertext Preprocessor) 25](#_Toc334205572)

[Вградени библиотеки 26](#_Toc334205573)

[Конфигуриране на PHP 27](#_Toc334205574)

[Предимства на PHP 27](#_Toc334205575)

[PHP и HTML 28](#_Toc334205576)

[Вграждане на PHP в HTML 28](#_Toc334205577)

[World Wide Web 29](#_Toc334205578)

Глава 3 Описание и реализация на проекта „Интернет портал за технологични новини”………………………………………………………………………………………36

[Предварително планиране 36](#_Toc334205579)

[Разработка на уеб сайта – стъпка по стъпка 38](#_Toc334205580)

[Структура на сайта 39](#_Toc334205581)

[Управление на съдържанието на сайта 44](#_Toc334205582)

[Защита на дизайна от счупване при произволно съдържание. 46](#_Toc334205583)

Заключение………………………………………………………………………………48

Приложение 1 - сорс кодове…………………………………………………………..49