

Szakdolgozat

Török Attila

Debreceni Egyetem
Informatikai Kar

Szerencsejáték portál és rulett szimulátor
megvalósítása

Témavezető: Dr. Nyakóné Dr. Juhász Katalin

Szakdolgozat készítője: Török Attila

Debrecen 2008. 12. 12.

Tartalomjegyzék

A honlapok készítésének szempontrendszere.....	6
A html nyelv rövid bemutatása.....	7
A javascript rövid leírása.....	10
Rövid php ismertető.....	13
Webszerverek.....	16
Apache webszerver:.....	17
Alkalmazás fejlesztés.....	21
Az index.htm.....	22
A szimulátorba történő beléptetés.....	24
A regisztráció folyamata:.....	26
Az adatbázis felépítése:.....	26
Az adatok ellenőrzése:.....	30
Link oldalak.....	31
A rulett.php oldal.....	37
A portál üzemeltetése, biztonsági kérdések.....	43
A portál üzemeltetése.....	44
A rendszer üzemeltetése.....	45
Összefoglalás.....	46
Felhasznált irodalom.....	48

Bevezetés

Szakedolgozatom témájaként a szerencsejátékok tárgyalását választottam. Napjainkban egyre népszerűbbek az internetes szerencsejátékok azon belül is talán a legnépszerűbb a rulett játék. A rulett látványos, gyors lefolyású, egyszerű, de nagyon szórakoztató játék. Ezen okokból egyre szélesebb tömegeket vonz. Magyarországon, egészen a legutóbbi időkig tiltott volt az emberek számára a kaszinók világa. A rendszerváltással, illetve a gazdaság liberalizálódásával ez a szegmens is betörni látszik az emberek életébe, egyfajta egyéb szórakozási alternatívát szolgáltatva a szerencsejátékokat kedvelő tömegek számára. Ma Magyarországon szigorú szabályok vonatkoznak a szerencsejátékok szervezésére, ezért szakedolgozatomban nem egy működő pénz szerzésére alkalmas alkalmazást készítettem, csupán arra szorítkoztam, hogy bemutassam a játék működését. Ezen szigorú szabályozások lazulni látszanak, hazánk európai uniós tagsága, valamint az internet terjedése, ezen belül a külföldi szerencsejáték portálok Magyarországról történő elérhetősége miatt.

Az elkészült honlap segítségével bárki kipróbálhatja minden kötelezettség nélkül, hogy mennyire szerencsés alkat, valamint esetleg kidolgozhat szisztémákat (megjegyzem, sem matematikailag sem pedig tapasztalataim alapján ennek értelme nincsen. Hangsúlyozom, hogy ez a magánvéleményem).

Szakedolgozatom témája egy honlap elkészítése, melynek központi eleme egy rulett szimulátor, ahol egy valós gépi rulett automata szimulációját valósítottam meg. A honlap foglalkozik a szerencsejátékok történetével, a kaszinókkal, a rulett játékszabályaival. Az egyik fontos témakör a szisztémák, matematikai esélyek latolgatása is. A honlap foglalkozik továbbá a szerencsejátékok veszélyeivel, élvezeti cikk révén könnyen szenvedélybeteggé válhat az, aki nem tart be néhány fontos szabályt.

A portál egyszerű felépítésű, áttekinthető struktúrájú, a hagyományos html elemeken túl alkalmazom a php, és a javascript nyelvi elemeit is. Maga a rulett szimulátor gyakorlatilag tisztán javascript nyelven íródott. Sajnos a javascript kód nem egységes, ezen okokból nem minden böngésző alkalmazás tudja az adott nyelvjárásban íródott kódot feldolgozni. Azon elképzelés, hogy meghatározott pozícióban egérekattintás esemény hatására megjelenhessen a kívánt tét a Mozilla Firefox böngésző képessége miatt lehetséges. A Mozilla böngésző egyfajta alternatívát biztosít az Internet Explorerrel szemben, felhasználói tábora egyre inkább nő. Jelen pillanatban a

böngészők versenyében, a második legnépszerűbb, részesedése 10-15%-körüli. Ez a részesedési arány több tíz millió felhasználót jelent világszerte. Magyarországon is egyre népszerűbb, ezért van létjogosultsága olyan honlapok kialakításának, amelyek támogatják a speciálisan ezen böngésző alkalmazásra írt javascript kódok megvalósítását. Természetesen a fejlesztés további lehetséges irányvonala az, hogy Internet Explorer kompatibilis program is készüljön, viszont erre, csak a forráskód teljes átdolgozásával van lehetőség. A másik inkább járható megoldás a java, vagy php nyelven történő kód megvalósítás, azaz szerver oldali alkalmazás készítése.

A szakdolgozat főbb témái:

- HTML nyelv rövid bemutatása
- Webszerverek
- Alkalmazás fejlesztés
- A portál üzemeltetése
- Összefoglalás

A honlapok készítésének szempontrendszere

A weblapok tervezésénél elsősorban törekedni kell a funkcionalitás a tartalom az esztétika összhangjára. A legfontosabb szempont, hogy a felhasználó amikor betöltődik a honlap, azonnal átláthassa a tartalmát lehetőség szerint azonnal, vagy pedig rövid ismerkedés után használatba vegye a honlap funkcióit.

Általánosan is érdemes megvizsgálni a honlapok tervezésének szempontjait.

Az alábbi felsorolás mutatja, hogy mik azok az ajánlott megoldások amiket szem előtt kell tartani.

- weblap kezdőlapja ne legyen Flash, DHTML, Java applet alapú
- A weblapon nem ajánlott frame-ek, iframe-ek használata
- Ne használjunk üres, átlátszó képeket design megoldásokhoz, pozícionáláshoz.
- Az oldalakon ne használjunk szerver oldali image-mapeket
- CSS formázó elemek nélkül is értelmezhető és használható legyen az oldal.
- Éljük a szabványok adta előnyökkel, fejlesszünk úgy, hogy az oldal szabványos legyen.

Tartalmak alternatív formában közzététele

- A navigáció nem épülhet nem hozzáférhető navigációs elemekre, mint Flash, Javascript

Zavaró elemek mellőzése

- Az oldal legyen 100%-ban(!) használható Javascript nélkül is. (ezt a szempontot nem tudjuk megvalósítani, a feladat jellegéből kifolyólag)
- Ne legyen háttérzeneje az oldalnak, ne legyen hang alapú visszajelzés az oldalon
- Formázások nélkül is használható legyen az oldal
- Az oldal linkjei ne nyissanak új ablakokat
- Az oldal háttere egyszínű legyen, ne tartalmazzon mintákat, képeket.
- Listázáshoz, tördeléshez, behúzáshoz az oldalban a megfelelő HTML elemeket használjuk
- Formázáshoz, díszítéshez karakteres szöveget (pl. >>) ne használjunk

Jól olvasható betűket használjunk

- A tartalmi rész előtt levő szöveg ne legyen túl nagy
- Az oldal szövegei értelmezhető blokkokra legyenek tördelve

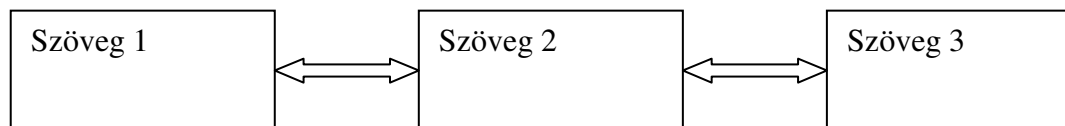
Meg kell jegyezni, hogy ezek nem szigorúan vett szabályok, elsődleges rendező elv a funkcionalitás és az esztétikum. A honlapok létrehozásánál egyfajta iránymutatást nyújtanak ezek a szabályok, de ettől a fejlesztő eltérhet a hatékonyság növelése érdekében. A fejlesztő számára a legfontosabb szempont a minél egyszerűbb megoldások alkalmazása, valamint a megrendelő és az alkalmazás helyes működésének szem előtt tartása.

A html nyelv rövid bemutatása

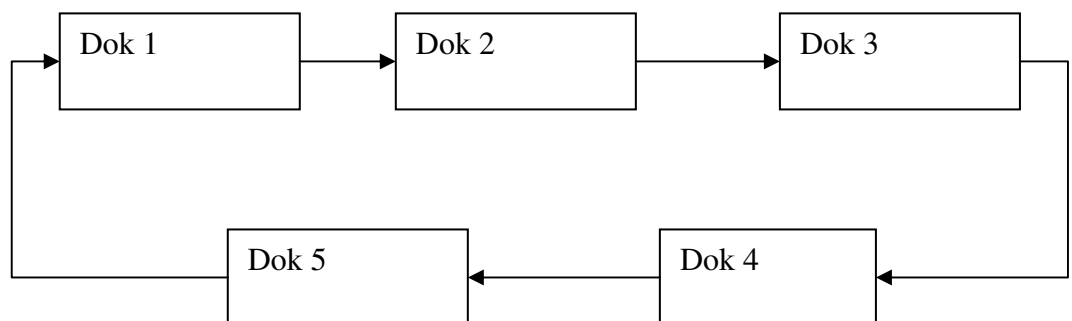
A honlapok, melyek információt jelenítenek meg, felfoghatók egyfajta olyan adatbázisnak amelyben az adatokat szöveg formátumban tárolják el (szövegszerű ismeretkezelés). Az egyes szövegrészeket, logikai egységeket pointereken azaz linkeken keresztül lehet elérni. Így a lapok egymáshoz való viszonya nagyon lényeges:

Lehet, oda vissza struktúra

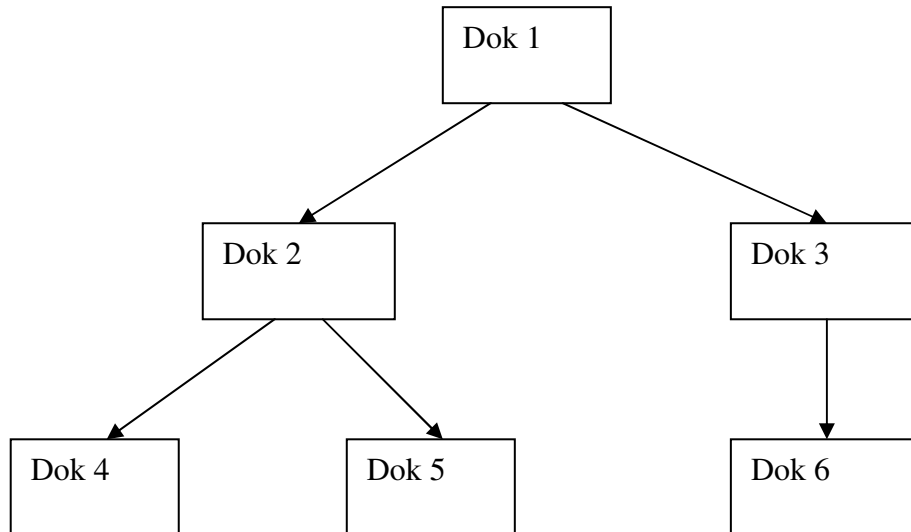
A linkek a dokumentumok között oda- vissza kapcsolatot valósítanak meg, egymásra mutatnak



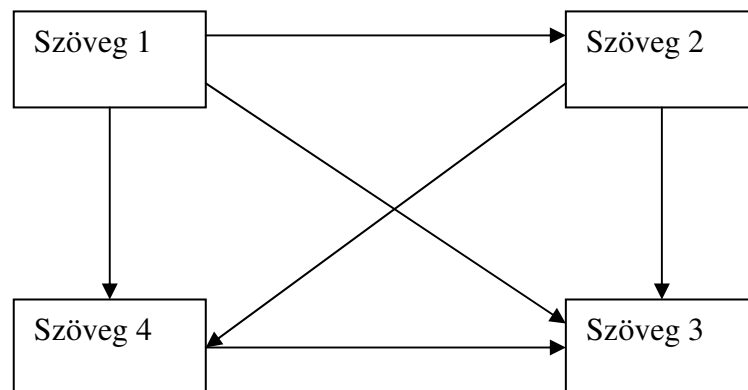
Körkörös: A dokumentumok egy meghatározott sorrendben követik egymást



Hierarchikus: A dokumentumok fa struktúrát alkotnak



Komplex: A dokumentumok egymást közvetlenül elérhetik, ilyen lesz az elkészítendő honlap is.



A honlapok információtartalmát tekintve lehet statikus, illetve dinamikus. Statikus honlapok adattartalma nem változik a használat során, a dinamikus honlapok esetén a felhasználó módosíthatja a honlap megjelenését. A kliens oldalról elküldött adatok függvényében rajzolja újra a böngésző a honlapot. A statikus honlapok esetén a honlap tartalma letöltődik a kliens számítógépére, majd a böngésző megjeleníti azt. Dinamikus honlapok esetén a szerver oldalon történik egy adatfeldolgozási folyamat amelynek eredményét adja át a statikus honlap számára a webszerver, majd ezután kerül át kliens gépre.

A HTML Hypertext Markup Language

A szövegszerű ismeretkezelés, annak adattartalmának a megvalósításának az eszköze. Egy html állomány tulajdonképpen egy szövegfájl, amelynek tartalmát formázó szimbólumokkal úgynevezett tag-ekel hozunk létre. Az állomány felépítése hasonlít a hagyományos harmadik generációs formális nyelvek tagolására. Eredeti verzióját a World Wide Web Consortium adta ki és a hivatalos szabványt ma is ez a szervezet jegyzi. Jelenleg a 4.01-es az utolsó elterjedett verzió.

Dokumentum típusdefiníció melynek lényege, hogy beállítjuk, hogy a dokumentumot a html melyik verziójában kívánja a fejlesztő megírni. Ezen bejegyzés determinálja a stílusok alkalmazásának módját illetve kompatibilitási megfontolásokat is felvet. Ez a rész elhagyható, ha nem specifikáljuk a DTD-t akkor az alapértelmezett szabvány szerint értelmezi a böngésző a honlap tartalmát, amely értelmében a szabványos stílusban jeleníti meg azt.

A fejrész amely ami technikai és dokumentációs adatokat tartalmaz, melyeket az internet böngésző nem jelenít meg, tehát átlag felhasználó ezeket nem látja.

A fejrészbe el lehet helyezni egyéb kliens oldali nyelvek alprogramjait, stílusok kialakításához szükséges eljárásokat.

A html törzs amelybe tulajdonképpen a dokumentum tartalmi része kerül formázó tag-ekel együtt.

A nyelvi elemek részletes ismertetésére terjedelmi okokból nincs lehetőség nincs is szükség rá, hiszen az Interneten számos oldal foglalkozik a nyelv részletes tárgyalásával. A formázó tag-eken kívül meg kell említeni az űrlapokat és azok elemeit, amelyekkel a felhasználó adatokat küldhet a szerver irányába. Az űrlap elemek nagyon hasonlítanak a negyedik generációs környezetek objektumkönyvtárában elhelyezett, gyárilag konstruált elemeire. Ezen elemek definiálása is hasonlít a vizuális környezetek megfelelő objektumaira.

A típus definíció gyakorlatilag egy értékadás, amely lehet text, password, submit, button, checkbox, radio image. Megadhatjuk az értékét az objektumnak, valamint a nevét, méretét. Meg kell jegyezni, hogy az űrlap elemek csak a formon belül létezhetnek.

A HTML nyelv egyik legmarkánsabb eleme a frame keret. A keretek segítségével a képernyőt felosztva egyszerre jeleníthetünk meg több webes dokumentumot a képernyőn. A frame-ek alkalmazása kissé háttérbe szorult, mivel hasznos lehet kiváltani használatát stílus elemek használatával.

A HTML nyelv nagy erőssége a képek elhelyezése a dokumentumban. Az elkészült honlap grafikus lehetőségeit nagymértékben emeli a kliensoldali térképek megvalósításának lehetősége.

Ne feldekezzünk meg azonban a legfontosabb képességről a linkek elhelyezésének lehetőségéről sem. A HTML nyelv nagyszerűsége abban rejlik, hogy keverhetők a kliensoldali nyelvek, szerveroldali nyelvek, amelyek a rendszer hatékonyságát nagymértékben növelik.

A javascript rövid leírása

Meg kell jegyezni, hogy a javascriptnek nincsen köze a java nyelvhez. A javascriptet először a Netscape böngészőkhöz fejlesztették ki, kiegészítve a HTML nyelv lehetőségeit. A nyelv egyik legnagyobb hátránya, hogy a nem egységes szabványok miatt, ugyanazon problémára különböző megoldásokat kínálnak. Ebből következően ugyanaz a kód két különböző böngészőben különbözőképpen hajtódik végre. A most bemutatásra kerülő javascriptben megírt rulett szimulátor program csak Mozilla böngészőben fut. Ezt a hátrányt a nyelv szabványosításával igyekeznek napjainkban kiküszöbölni. A másik hátrány, hogy nem készült a nyelvhez hibakereső. Ez a probléma az alkalmazások fejlesztési idejét megnöveli, illetve körültekintő rendszertervezési és programozási feladatot igényel a fejlesztőktől. A nyelv felépítése hasonlatos az integrált fejlesztői környezettel rendelkező formális nyelvekhez. Nyelvi elemei szintaxisát tekintve a C nyelvhez hasonlíthatnak. A javascript képességeit nagymértékben megnöveli az objektumok használatának lehetősége.

A javascript parancsértelmezője interpreterje a struktúrákat soronként fordítja illetve a struktúrákat a pontosvesszőig értelmezi szekvenciálisan.

A nyelv beilleszethezősége miatt html tag-ek közé helyezik el a script uatsításait

```
<script type="text/javascript">
```

```
*
```

```
*
```

```
Javascript kódok
```

```
*
```

```
*
```

```
</script>
```

A scriptet elhelyezhetjük a honlap Head és a Body szekciójában, valamint meghívható külső forrásként is:

```
<head>  
<script src="xxx.js"></script>  
</head>
```

A nyelvi elemek:változó deklarációk, értékadás, elágazások ciklusok, függvények I/O szerkezetek megtalálhatók a dolgozat mellékletében.

Fontos megemlíteni viszont a nyelv objektum orientált képességeit.

A javascriptben lehetőség van előre beépített objektumok alkalmazásán kívül a fejlesztő által definiált objektumok létrehozására is.

Az alábbi példák alapján látható hogy a javascriptben használt tulajdonságok és metódusok alkalmazása megegyezik bármely magas szintű programozási nyelvben használt formával

```
<script type="text/javascript">  
var txt="helló Világ!"  
document.write(txt.length)  
</script>
```

Látható, hogy a txt string objektum length tulajdonságát adja vissza a document objektum write metódusa amelynek eredménye 10 lesz, hiszen a szöveg 10 karakterből áll.

Ha saját objektumot szeretnénk megtenni azt a következő formában tehetjük meg:

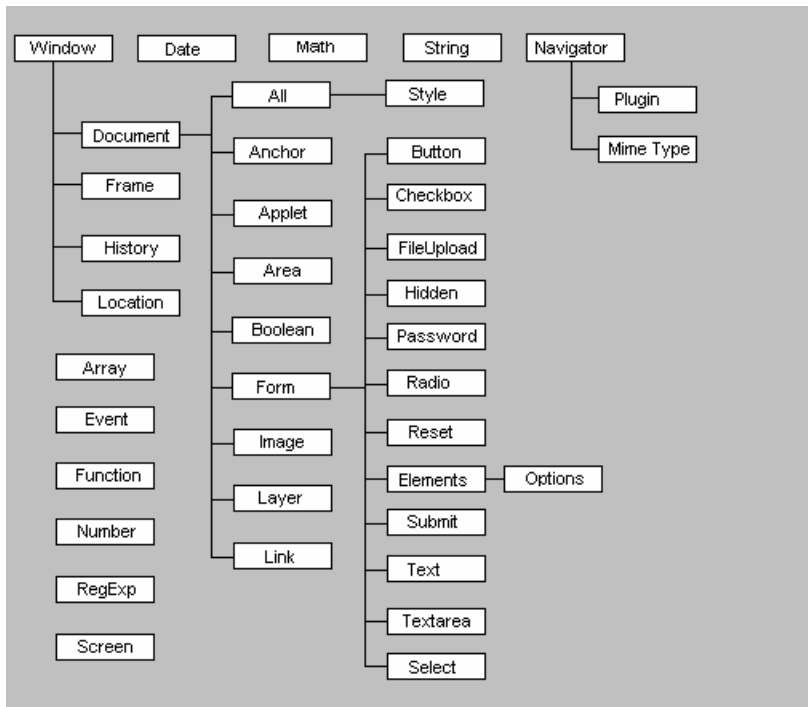
```
szemelyObj=new Object()  
szemelyObj.keresztnev="Attila"  
szemelyObj.veznev="Török"  
szemelyObj.kor=34  
szemelyObj.szemszín="barna"
```

A „szemelyobj” nevű objektumot először létrehozuk a new Object() objektumból, majd felruházzuk saját magunk által definiált tulajdonságokkal. A tulajdonságoknak pedig, értéket adunk.

JavaScript Objektumok Listája

All A HTML fájl dinamikus programozása **Anchor** Cimkék (hivatkozási pontok) **Applet** Java Applettek beemelése **Area** Grafikus hivatkozási terület (link) **Array** Tömb objektum **Boolean** Boolean objektum **Button** Grafikus nyomógomb az űrlapon **Checkbox** Csekkbox az űrlapon **Date** Dátum és idő **Document** Az ablakban lévő dokumentum **Element** Űrlapok elemei **Event** Események figyelése **FileUpload** Fájl feltöltés a szerverre **Frame** Több ablak a képrnyőn **Form** Űrlap megjelenítése **Function** Függvények tulajdonságai **Hidden** Láthatatlan elem az űrlapon **History** Meglátogatott oldalak **Image** Grafikák **Layer** Rétegek kezelése **Link** Linkek (kapcsolatok) **Location** URL cím kijelzése **Math** Matematikai állandók és függvények **MimeType** MimeType információk **Navigator** Böngésző tulajdonságok **Number** Numerikus értékek **Option** Választólista **Password** Jelszó beviteli mező **Plugin** Beépülő programok **Radio** Rádió gomb **RegExp** Illesztő kifejezések **Reset** Törlés gomb **Screen** Képernyő információk **Select** Választólista **String** Karakterláncok kezelése **Style** Stíuselemek megadása **Submit** Elküldi az űrlap adatait **Text** Egysoros szövegbeviteli mező **Textarea** Többsoros szövegbeviteli mező **Window** Megjelenítő ablak

Valamint a javascript objektumok hierarchiája



Rövid php ismertető

A PHP nyelv nem olyan régi mint a C nyelv, eredete visszavezethető 1994-ig amikor is Rasmus Lerdorf elkészítette a rendszer első verzióját saját munkájának megkönnyítésére. A nyelvet speciálisan dinamikus weboldalak készítésére használják. A rendszer nagy előnye, hogy együttműködik a mysql adatbázisokkal így összetett adatkezelő rendszerek is kialakíthatóak web alapú környezetben. Napjainkra nagyon elterjedté vált az egész világon, jelenleg a nyelv 5. verziója a legújabb.

Ha a PHP szintaktikáját megnézzük az csakúgy mint a javascript nagyon hasonlít a C nyelvre. A php fordító számára a struktúra végét pontosvessző jelzi. A php kódot tartalmazó állományok kiterjesztése „.php” A kódban a PHP HTML elemek vegyesen alkalmazhatóak. A php kódok a html dokumentumban <?php ?> tag-ek közé kerülnek. A html kódok elérése php ből a print függvénybe elhelyezendő tag-kel lehetséges. Megjegyzem, ugyanígy lehet javascript rutinokat is beszúrni php állományba. A php csakúgy mint a legtöbb magasszintű nyelv a nyelvi elemeken (I/O szerkezet, értékadás, ciklusok, elágazások, alprogramok) kívül rengeteg beépített függvényel rendelkezik. A PHP nyelv beépített függvényeit az alábbi függvénycsoportokba ehet sorolni:

- Adatbázis-kezelők elérése (MS-SQL, MySQL, Oracle, SyBase, PostgreSQL, dBase, filePro, Informix, InterBase, mSQL, dbm, ODBC)
- tömbkezelés,
- matematikai függvények,
- file-kezelés,
- programfuttatás,
- képlétrehozás/manipulálás,
- IMAP, LDAP
- Kódolás
- Hálózat
- NIS
- PDF
- string
- url
- tömörítés
- XML.

A php gyengén típusos nyelv ellentétben például a Pascallal mert a változó-típusok és a változók egyeztetése (pl. értékadásnál) automatikus konverzióval történik. Mivel a PHP-ben írt programok tulajdonképpen CGI-k ezért különösen érdekes a web szerver által átadott változók elérése. Háromféle külső változó típust különböztet meg a PHP, ezek:

1. GET metódus által kapott adatok
2. ugyanez POST metódus esetén
3. HTTP Cookie változók.

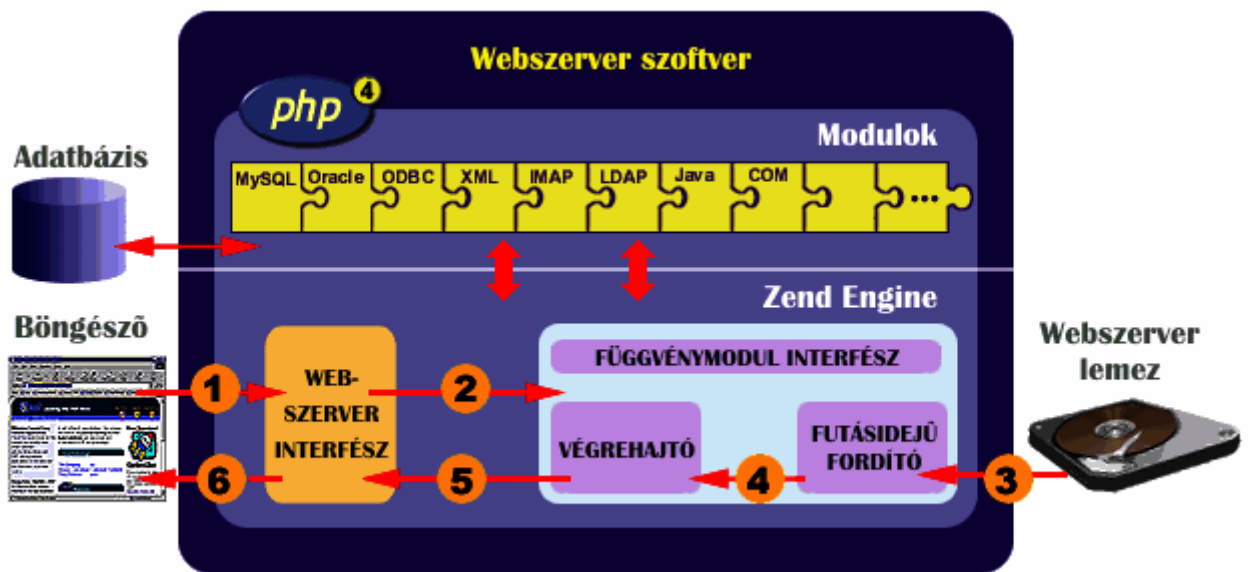
(Ha fenti három pont értelmezésében probléma adódna, akkor ajánlom az olvasó figyelmébe a párhuzamosan futó Kommunikáció rovat előző két számát, mivel abban a CGI-ről volt szó.)

A kívülről kapott paraméterek kezelése nagyon egyszerűvé válik azáltal, hogy a PHP minden egyes paramétert automatikusan átalakítja a programban használható

változókká. Például ha egy HTML form-ban a kitöltendő mező neve: cím, akkor a script meghívásakor ebből egy \$cím nevű globális változó lesz (a \$-el a változókat jelöli a nyelv a Perl-hez hasonlóan), amit ugyanúgy használhatunk, mint bármely más programváltozót. Néha persze kavardást okozhat, hogy több helyről is kaphatunk változókat akár ugyanazzal a névvel is (GET/POST/Cookie), ezért felállíthatunk egy tetszőleges prioritást, vagy akár le is tilthatjuk az egyik forrás automatikus átalakítását.

Négy féle változótípust használhatunk a programjainkban: egész szám, valós szám, string, tömb, objektum. Az utóbbiból kiderül, hogy a PHP támogatja objektumok/osztályok létrehozását. Azért nem kell mindjárt Java-szintű objektumorientáltságra gondolni, a PHP-ben az objektumokat elsősorban könyvtárak írásakor használják, mivel így az objektumokba zárt változók nem érintik a globális társaikat. A PHP az objektumokra vonatkozó zártságon kívül a konstruktorok használatát és az egy szülőtől való öröklést teszi lehetővé. Jól hasznosíthatóak az objektumok például adatbázis-szerverek absztrakt kezelésére, elősegítve ezzel a programjaink hordozhatóságát (a már említett PHPLib is objektumokat használ erre a célra).

A tömbkezelés is a PHP egy nagyon fontos része, mivel az adatbázisból nyert adatainkat általában egy vektorban kapjuk vissza, ezt annak köszönhetően tehetjük meg, hogy a vektorokban nem csak azonos típusú elemeket tárolhatunk. Ezen kívül lehetőségünk van HTML űrlapokban elhelyezett adatokat is vektorként megkapni. Ez hasznos lehet például akkor, ha kiválasztó lista, vagy checkbox-ok adatait szeretnénk visszanyerni, mert vektorok használatával egyszerűbben végig tudunk haladni a kiválasztott elemeken. Az asszociatív tömbök használatával szintén könnyebbé tehetjük életünket (és persze olvashatóbbá a kódunkat), mivel sokkal egyszerűbb az adatbázis egy rekordját tároló tömbből, úgy kiválasztani a megfelelő elemet, hogy nem kell emlékeznünk, hogy "hányadik a sorban", hanem egyszerűen a mezőnévvel hivatkozhatunk rá, pl.: személy["név"]. A fentieket megfejelve még számos (34) függvénnyel szolgál a PHP a tömbök igazán hasznossá tételéért, pl.: rendezés (véletlenszerűvé is), elem beszúrás, verem függvények (push, pop), elemcsere, elemeken való lépkedés...



Az ábra alapján érdemes a php script feldolgozás folyamatát megvizsgálni:

Az első lépés a kapcsolat felépítése a webszerver irányába.

A webszerver php modulja kialakítja a szkript futtatásához szükséges környezetet, majd ezután az online fordító a php nyelv szintaxisának megfelelően lefordítja a rutint és outputot generál, amelyet a webszerver továbbítja tcp/ip protokollon keresztül a böngészőnek a kódot.

Webszerverek

Jelenleg két PC alapú webszerver alkalmazás terjedt el. A linux alapú környezetben az Apache, a windows alapú rendszerek esetén pedig az IIS.

természetesen vannak törekvések arra is, hogy Apache webkiszolgálót telepíthessünk windows operációsrendszerekre is. A legkényelmesebb megoldást az Appserv alkalmazás telepítése nyújtja, amely bárki számára elérhető és letölthető az Internetről. A telepítő program letöltése után az installáció folyamata automatizált. Az Appserv programcsomag tartalmazza az Apache webkiszolgálót, PHP fejlesztői környezetet, MySQL szerveret.

A telepítés után a rendszer automatikus alapbeállításokkal kerül elindításra amely azonnal használhatóvá válik. Természetesen a megfelelő konfigurációs állományokkal a rendszer a felhasználói igényeknek megfelelően testre szabható.

A továbbiakban az Apache webservert röviden ismertetésre.

Apache webservert:

Az alkalmazás jelenleg az összes szerverek 80%-án használják. Elterjedtségének okait a következőkben kell keresni:

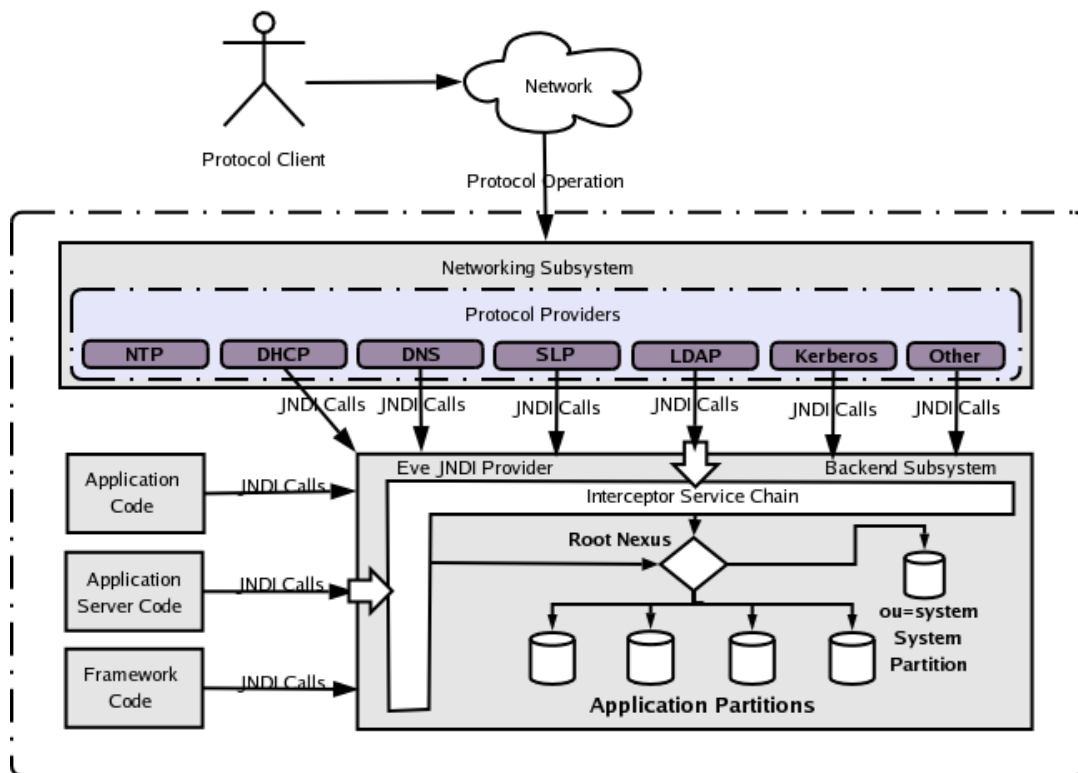
- nyílt forrás
- modularitás
- portolhatóság
- nagy teljesítmény
- megbízhatóság
- biztonság

Legfontosabb konfigurációs fájl a *httpd.conf*, állomány. Lássuk a legfontosabb paramétereket:

A *Port* - az alapértelmezett hozzáférési port száma, ez átlagos esetben 80. *User www-data Group www-data* - az Apache-ot futtató user, illetve csoport azonosítója *ServerAdmin* - a webservert gazdájának e-mail címe *ServerName* - amennyiben a szervernév eltér a hosztnévtől, itt beállíthatjuk azt *DocumentRoot* - a szolgáltatott webtartalom gyökerének elérési útja

A fenti paraméterek beállításával egy átlagos környezetben használt webservert működése máris megindulhat. Következő feladatunk a kiszolgálandó tartalom elhelyezése, melynek a */www/* könyvtár ad otthont, de természetesen bárhol elhelyezhetjük, mindössze arra ügyeljünk, hogy a webservert futtató usernek legyen hozzá olvasási joga és módosítsuk a *DocumentRoot* paramétert. A szerver működése során rendkívül fontos a naplófile-ok megfigyelése, melyeket legtöbb esetben a. Az **access.log** a lekért tartalmakat, míg az **error.log** a hibákat gyűjti össze.

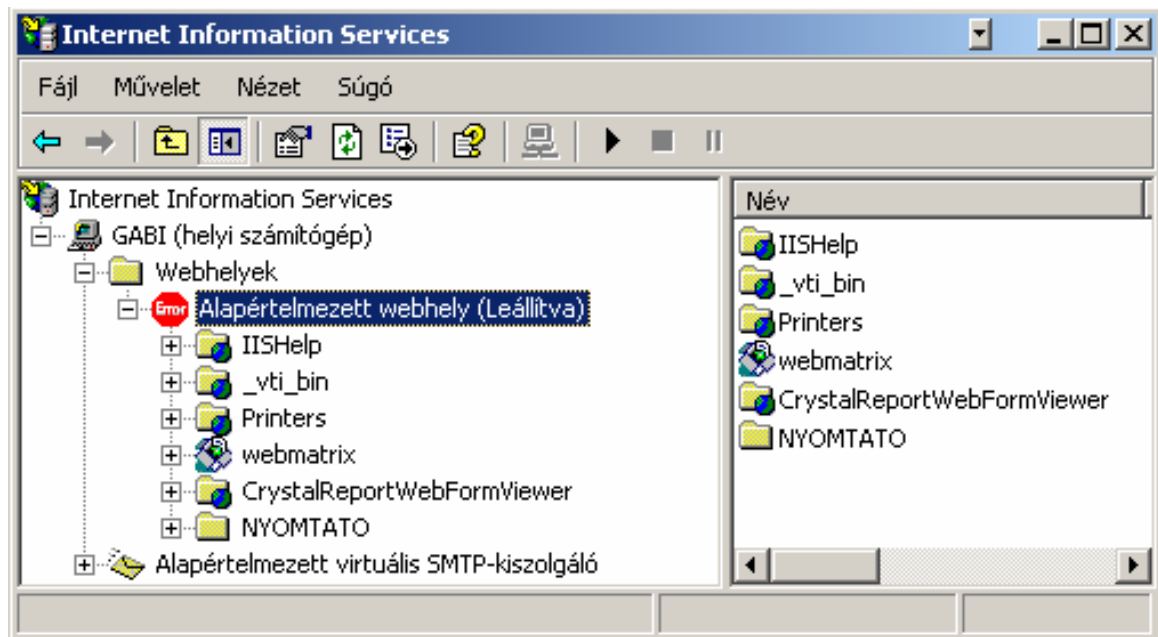
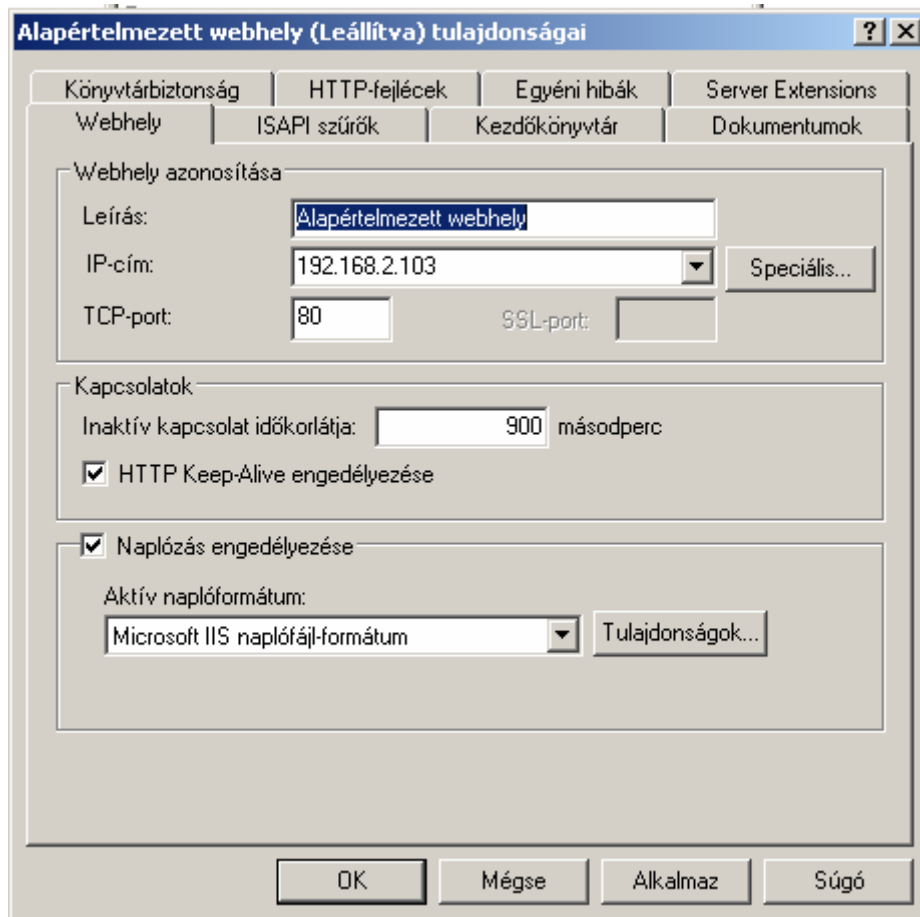
Az alábbi angol nyelvű ábrán a webservert architektúrája látható:



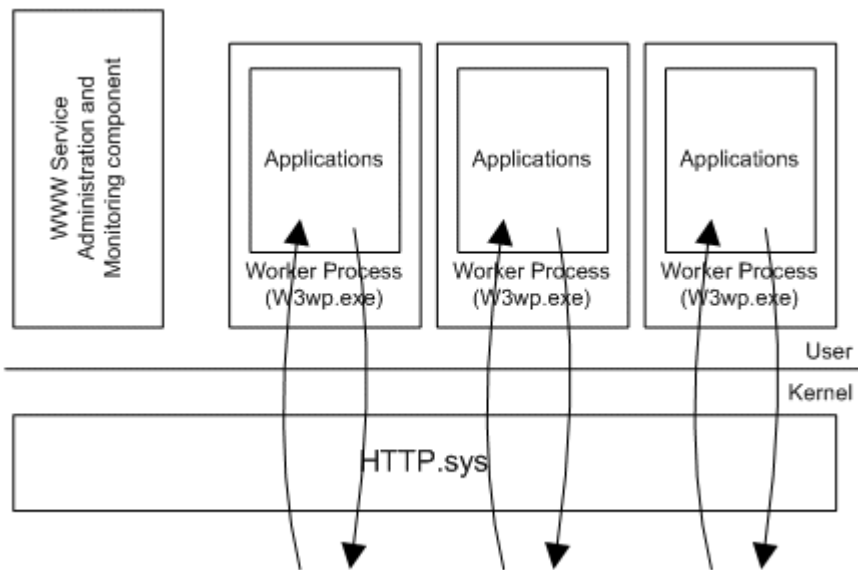
Az ábrán látszik a webszerver moduláris felépítése

Az Internet Information services IIS webkiszolgáló

Az IIS webszerver alkalmazás a Microsoft fejlesztése, opcionálisan telepíthető a Microsoft XP operációs rendszerre. Az alábbi ábrákon is látszik, hogy nem parancsállományok segítségével lehet konfigurálni a webszervert, hanem dialógus ablakok segítségével, amely nagy mértékben megkönnyíti a felhasználó dolgát, hiszen egyszerűbb és átláthatóbb kezelő felületek segítségével hajtható végre a konfiguráció



Az IIS architektúrája:



a webserverek a dinamikus weboldalak előállító programok futtatását az ún. Common Gateway Interface (CGI) szabványban rögzített felületen keresztül teszik lehetővé. Ennek lényege, hogy a dokumentumletöltési kérés beérkezése esetén a kiszolgáló egy külső futtatható állományt hív meg, amelynek átadja a kérés paramétereit, és amelytől a válaszként a böngésző felé továbbítandó dokumentumot kapja vissza.

Alkalmazás fejlesztés

A szerencsejáték portál felépítése „hagyományos” áttekinthető szerkezetű. A honlap bejelentkező képernyője (index.htm) tartalmaz minden elérhető funkciót, innen lehetséges belépni a rulett szimulátorba, valamint minden link és letöltés erről az oldalról navigálható. Az oldal felépítése hagyományos hármastagolású, mind vertikális mind pedig horizontális irányban. Funkcionálisan jól elkülöníthető részekből áll. Az egyszerű struktúra mind kezelhetőség, mind funkcionalitás szempontjából előny, hiszen akár gyakorlatlan felhasználók is könnyen eligazodhatnak a tartalomban. Ha horizontálisan vizsgáljuk az oldal struktúráját, a felső és mondhatjuk központi elem, a rulett szimulátorba történő belépés és adminisztrációs részt jeleníti meg. Az oldal középső részén egy menüsor található, amely egyszerű nyomógombokkal lettek kialakítva. A menüsorban lettek elhelyezve azok a funkciók melyek a legfontosabb információkat tartalmazzák a honlap működésével kapcsolatosan. A harmadik rész a tartalmi rész amely vízszintesen két, függőlegesen pedig három egységre lett szétosztva. A tartalmi egység első sora gyakorlatilag a portál tartalmának központi eleme ahol tulajdonképpen az írásos, információs anyagot találjuk. A második sor pedig a honlap járulékos funkcióit tartalmazza, azaz linkeket és kiegészítő funkciókat, mint például linket a Mozilla letöltésére, ha valaki esetleg szeretné elérni a rulett szimulátort és a fenti böngésző programmal akarja megjeleníteni a portál tartalmát. Megjegyzem, hogy a honlap természetesen olvasható bármilyen programmal, de a rulett szimulátor, csak ezzel a böngésző programmal, működik kifogástalanul.

A portál belső szerkezetét tekintve három fő laptípusból áll össze. Az index.htm oldalból, amely egyedi szerkezetű, valamint a linket és a tartalmat hordozó oldalakból, amelyek egységes stílusúak és szerkezetűek, elősegítve ezzel az áttekinthetőséget és előtérbe helyezve a funkcionalitást. A harmadik egység maga a szimulátor programot tartalmazó oldal, amely gyakorlatilag a kliensoldali javascript kódot tartalmazza. Elmondható, mivel a honlap elemek gyakorlatilag rögzített pozícióba lettek elhelyezve, bármely felbontásban kiválóan működik. Ajánlott felbontás 1024*768, viszont a 800*600-as képernyőfelbontásban is élvezhető a portál megjelenése.

Az index.htm

Ebből az oldalból egyetlen példány található a portálon. Felépítése teljes mértékben egyedi. Ez a szerencsejátékokkal foglalkozó portál kezdő oldala.



Roulette

lépjen be a rulett szimulátorba!

Felhasználói név :

Jelszó:

Ha nincs felhasználói jelszava, kérem kattintson a regisztráció gombra!

[A szerencséről](#) [Játékszabályok](#) [A szimulátor](#) [Honlaptérkép](#) [Kapcsolat](#)

<p>A szerencsejátékok veszélyei Milyen hatásai vannak a szerencsejátékoknak? Hogyan lehet elkerülni a szenvedély betegséget?</p> <p>Tovább</p>	<p>Hogyan lehet nyerni? Lehet-e egyáltalán nyerni a ruletten Elemzések Matematikai valószínűségek</p> <p>Tovább</p>	<p>Érdekes történetek Életről vett sztorik, Kitalált mesék Cikkek Tudjon meg többet a szerencsejátékosok életéről</p> <p>Tovább</p>
<p>Hasznos linkek Rulett lap Szerencsejáték lap Sporting bet Index fóruma Szerencsejáték zrt</p>	<p>Egyéb szerencsejátékok Black jack Nyerő automaták</p> <hr/> <p>Mozilla letöltés</p>	<p>Kaszinók lapjai Casinoacademia Amerikaipóker Várkert kaszinó</p>

A bejelentkező oldal kilenc jól meghatározott egységre tagolódik, ezeket „Div” html tag-kel oldottam meg. A „Div” tag-ekkel történő kialakítás előnye, hogy a tartalom könnyen pozícionálható, így sokkal nagyobb rugalmasságot lehet megvalósítani az

oldalon belül. Az így kialakított blokkokba könnyen elhelyezhető a tartalom. Jelen esetben táblázat egy celláját helyeztem el a konténerekbe, pusztán azon megfontolásból, hogy így egyszerűen keretet rajzoltam a konténer köré.

A következő kódrészlet alapján látható hogyan került kialakításra az oldal struktúrája:

```
<div id="Kaszinók" style="position:absolute;background-color:#ccff99;
top:515;left:419">
<table border="3" cellpadding="0" cellspacing="0" style="border-collapse: collapse"
bordercolor="#111111" width="210" height="130" id="AutoNumber3">
  <tr>
    <td width="200" height="130">
      <p align="center" >
        <font size="4" > <b> Kaszinók lapjai
        <br>
        <A HREF="www.casinoacademia.hu"> Casinoacademia</a>
        <br>
        <A HREF="http://www.amerikaipoker.hu/"> Amerikaipóker</a>
        <br>
        <A HREF="http://www.varkert.com/"> Várkert kaszinó</a>
        <br>
        </b>
      </font>
    </p>
  </td>
</tr>
</table>
</div>
```

A div blokk stílus attribútumának paraméterei segítségével pontosan végrehajtható a lapon történő pozicionálás, valamint a blokk mérete, színe. A blokkban elhelyezett tábla, amely tulajdonképpen egyetlen cellát tartalmaz, illeszkedik a konténer széleire. A blokkon belül található a tartalom. Jelen esetben „hagyományos” linkek, amelyek egy bekezdés segítségével és font tag-gel lettek megformázva.

Az index.htm szöveges tartalomhoz kapcsolódó szegmenséből egy „button” objektumra történő kattintással léphetünk át másik lapra.

```
<input type="button" style="background-color:#0066cc;cursor: pointer" value="Tovább" name="B1" onclick="document.location.href ='tortenetek.htm'">
```

A fenti kódrészlet alapján látható, hogy a nyomógomb stílusa a következőképpen változott meg. A gomb háttérszíne kékárnyalatúvá vált , az egérmutató megváltozik (pointer) ha a kurzor a gomb fölé érkezik. Egér kattintás esemény bekövetkezésekor a document.location.href tulajdonság megkapja a linkelt oldal értékét.

A menüsor a fent leírt módon, azaz nyomógombok segítségével lett kialakítva. Minden egyes nyomógombra történő pozicionálás esetén button objektum style attribútum curzor paraméterei pointer értéket kapnak, valamint az egér kattintás esemény hatására a megfelelő link megnyitásra kerül. A kapcsolat gombra kattintással viszont az alábbi kódrészlet alapján az alapértelmezett levelező program nyílik meg:

```
onclick="location.href='mailto:abc@freemail.hu'"
```

A szimulátorba történő beléptetés

A szimulátorba, egy jogosultsági rendszeren keresztül lehetséges belépni. Ennek oka az, hogy minimális információt kapjunk azon felhasználókról, akik szeretnének játszani a honlap rulett játékán.

A jogosultság megszerzése a regisztrációs űrlap kitöltésével történik. A regisztráció után a főoldalon a beviteli űrlap mezőiben meg kell adni a felhasználói nevet és a jelszót. Helyes felhasználói név és jelszó megadása esetén elérhetővé válik a szimulátor kezelőfelülete.

A regisztráció folyamata:

A regisztrációs oldal tulajdonképpen egy űrlap, amelynek kitöltésével és elküldésével az adatok bekerülnek a mysql adatbázisba.



Név:

Cím:

e-mail cím:

Felhasználónév (kötelező kitölteni!):

Jelszó (kötelező kitölteni!):

Jelszó megegyeszer (kötelező kitölteni!):

Az oldal forráskódja három logikai egységből épül fel. A később ismertetésre kerülő a lap megjelenését alkotó vázból, magából az űrlapból amely tulajdonképpen egy form ami saját magára hivatkozik az alábbi kód alapján :

```
<form action = "<?php echo $PHP_SELF?>">
```

A harmadik egység a php-ban íródott kód, amely beírja az űrlap adatait az adatbázisba.

Az adatbázis felépítése:

Jelen pillanatban nagyon egyszerű felépítésű adatbázist használ a rendszer. Nincs szükség jelen pillanatban sok adatra, hiszen az adatbázis nem marketing célokat szolgál, valamint nincs semmilyen adózási és jogi kötelezettsége a honlap üzemeltetőjének, sem a hatóságok, sem pedig a felhasználók felé. Ezen okok miatt az adatbázisba a név, cím,

e-mail cím felhasználói név és a jelszó kerül. Nem kötelező kitölteni csak a felhasználói nevet. Ennek oka az, hogy csak a felhasználói név alapján lehet megkülönböztetni a felhasználókat. Jelen pillanatban a rendszer nem naplózza a belépéseket, nem gyűjt semmilyen információt. Ez lehet a rendszer tovább fejlesztésének lehetséges iránya, de a portál elsődleges célja a felhőtlen szórakoztatás mindenfajta kötelezettség nélkül. Adatbázis tervünk jelenlegi állapotában nem szükséges a táblát normalizálni (megjegyzem, hogy a név és a cím tulajdonságok nem atomi attribútumok, amelyeket, ha szükséges a későbbiekben az adattábla dekompozícióját hozhatja, és normalizálással az anomáliák megszüntethetőek.)

A fenti megállapításokat követve a logikai modell egyedtípus szerkezeti listája egyetlen funkciódefiníciós lapra szorítkozik:

Statikus modell	Egyedtípus szerkezeti lista	Lapszám: 1
Dokumentum kód: 1		Készítette: Török Attila
Változatszám :1		Dátum 2006. április 30
Egyedtípus azonosítója és neve: adatok		
Tulajdonság típus neve:	Tulajdonságtípus szöveges értelmezése	Típusa
nev	A felhasználó neve	leíró
cim	A felhasználó lakcíme	leíró
e-mail	A felhasználó e-mail címe	leíró
felhaszn	Az egyedi felhasználói név, becenév	azonosító
jelszo	A felhasználó jelszava	leíró

Az adatbázis fizikai modellje egy ilyen táblázat:

Statikus modell	Fizikai modell		Lapszám: 1
Dokumentum kód: 1			Készítette: Török Attila
Változatszám :1			Dátum 2006. április 30
Tábla azonosítója és neve: adatok			
Mező neve:	Mező típusa	Mező hossza	Megjegyzés
nev	varchar	50	
cimm	varchar	50	
e-mail	varchar	50	
felhaszn	varchar	10	Kulcs, index
jelszo	varchar	10	

Az adatbázis adattartalmát csupán egyetlen esemény befolyásolja az pedig nem más, mint a regisztráció művelete, azaz új felhasználó felvitele az adat táblába.

Az adatbázishoz történő csatlakozás és az adattábla kiválasztása után következik a konkrét adatbázis-kezelési művelet, amelynek legfontosabb eleme nem más mint egy SQL lekérdezés.

```
$eredmeny = mysql_query("select * from adatok where felhaszn='$felhaszn'");
```

```
$sorok_szama = mysql_num_rows($eredmeny);
```

```
If(!empty($felhaszn))
```

```
{
```

```
if($sorok_szama==0)
```

```
{
```

```
$ujsor="Insert into adatok(nev,cim,email,felhaszn,jelszo)
```

```
values('$nev','$cim','$email','$felhaszn','$jelszo');
```

```

        if(!mysql_query($ujsor,$kapcsolat))
        {
            print("<br> hiba");
        }

        else
        {
            echo "<script>alert('A regisztráció sikerült!');</script>";
        }

    }
else

    {

        echo "<script>alert('Létező felhasználó!');</script>";

    }

}

```

Mivel a felhasználói név az adattáblában elsődleges kulcsként van definiálva – ennek oka, hogy nem lehetséges két azonos felhasználó megléte- fontos annak vizsgálata, hogy létezik-e a megadott néven felhasználó vagy sem. Ha igen, akkor valamilyen visszajelzést kell adni, amit a program meg is tesz javascript üzenet ablak formájában. Ez az oka annak, hogy az űrlap állomány nem különül el a php kódot tartalmazó állománytól, hanem egy oldal tartalmazza mindkét kódot. Egy egyszerű „Select” utasítás segítségével vizsgálja meg a program, hogy létezik –e a meghatározott néven felhasználó. Ha létezik akkor természetesen egy rekorddal tér vissza az SQL lekérdezés eredmény halmazára. Ha nem szerepel ilyen néven felhasználó, akkor a lekérdezés üres halmazt ad vissza eredményül. Annak az eldöntése, hogy van-e visszatérési érték vagy sem egy if-es szerkezet segítségével, valósítható meg. Az adatok beírása az adattáblába paraméteres lekérdezés segítségével történik, ahol a paraméterek értékeit az

űrlapváltozók által átadott adatok képviselik. Sikeres, vagy sikertelen adatbázis művelet végrehajtásról a javascriptben megvalósított üzenetablak ad információt. A fenti kódban keverednek a php, a html és a javascript elemek. Ez is azt mutatja hogy, nagyon hatékony és rugalmas kódok valósíthatók meg a különféle internetes technológiák együttes alkalmazásával.

Meg kell jegyezni, hogy további fejlesztést kíván az űrlap kitöltés- szabályrendszerének és validitásának vizsgálata. A program jelen pillanatban nem vizsgálja minden beviteli mező kitöltöttségét, és a jelszó egyezést. Ezen hiányosságok javítása a rendszer továbbfejlesztésénél elsődleges feladatnak tekintendő.

Az adatok ellenőrzése:

Az adatok ellenőrzését az „ellenoriz.php” állomány hajtja végre.

Az alábbi kódrészlet alapján látható, hogy a program megvizsgálja a felhasználói nevet és a jelszót. Ha az elküldött adatok egyszerre, azaz egyetlen rekordon belül léteznek, akkor beengedi a játékost a rulett szimulátorba, tehát az SQL lekérdezés feltételében megadott paraméterek alapján egy rekorddal kell visszatérni a sikeres belépéshez. A kódrészletben az is látható, hogy sikeres belépésnél indul egy session, amely adatbiztonsági szempontból lényeges momentum.

Ha a fenti feltétel nem valósul meg, akkor marad az eredeti index.htm állomány megnyitva.

```
$eredmeny = mysql_query("select * from adatok where felhaszn='$felhaszn'and  
jelszo='$jelszo'");
```

```
$sorok_szama = mysql_num_rows($eredmeny);  
/*print("<br> a sorok száma:$sorok_szama");*/
```

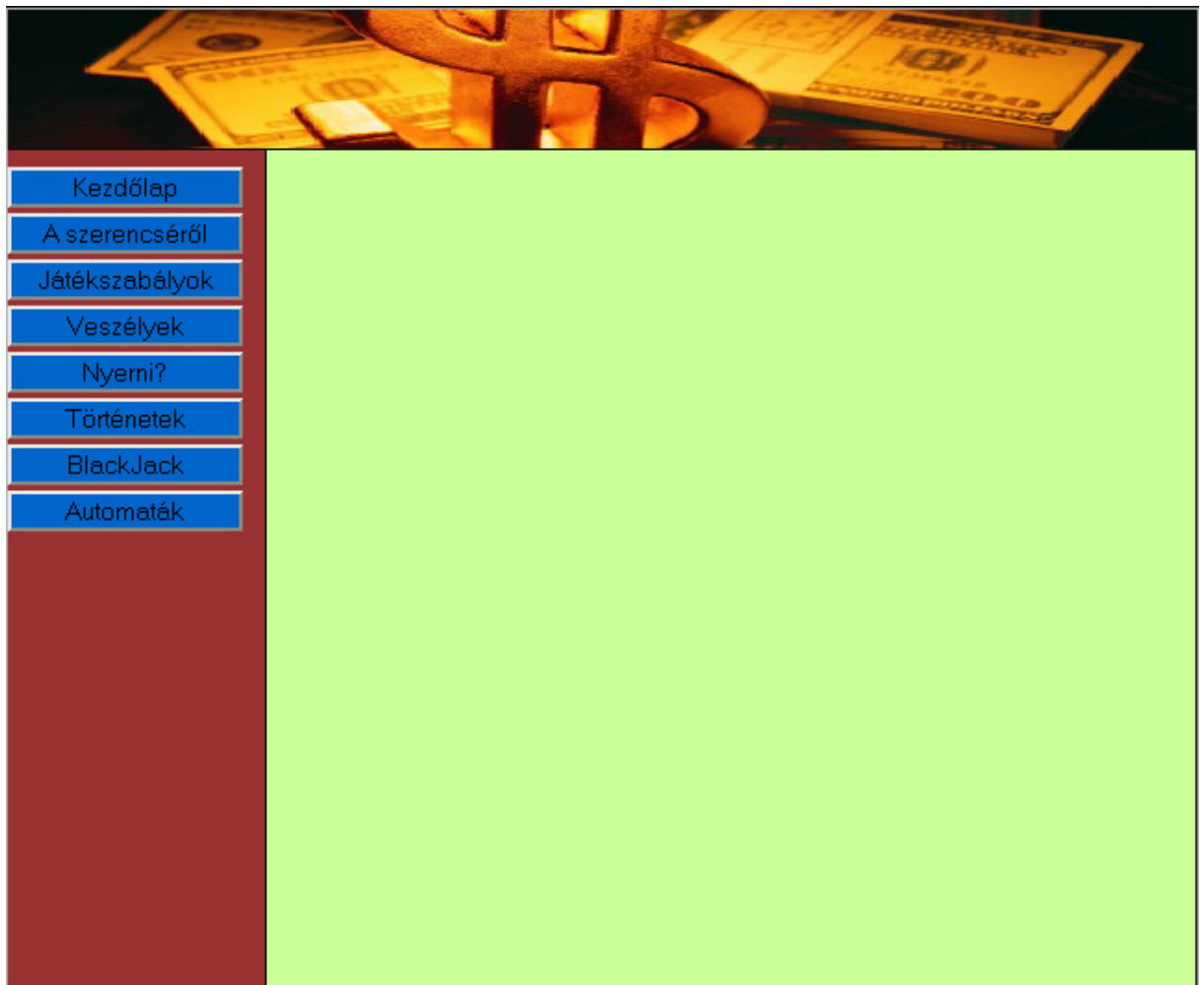
```
if($sorok_szama >=1)  
{  
    session_start();
```

```
$_SESSION["auth"]=1;
header("Location: rulett.php");
exit;
}
else
{
header("location: index.htm");
exit;
}
```

Link oldalak

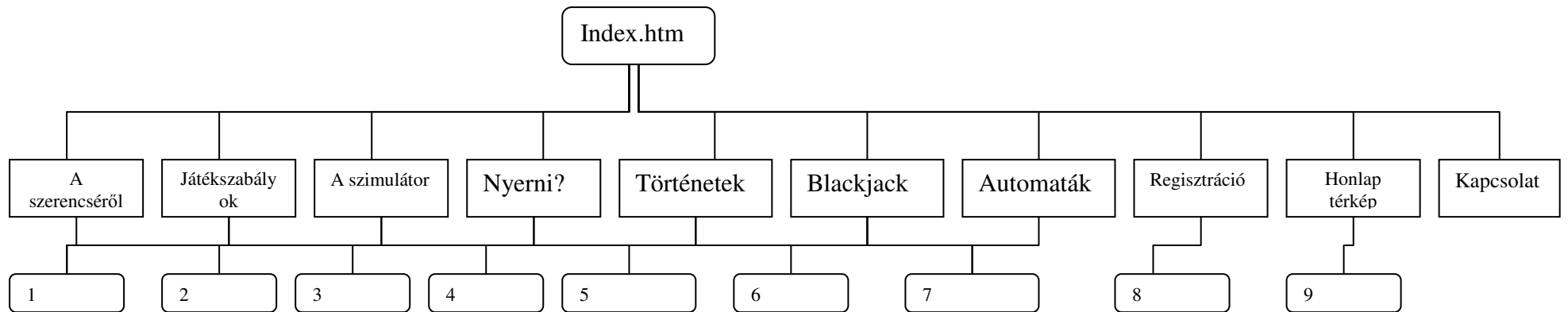
Az index.htm lapról elérhető oldalak stílusa és felépítése egységes képet mutat.

Minden linknek a felső szegmensében egy kép található, a bal oldalon az elérhető funkciókat tartalmazó menüsor került elhelyezésre, amelyek gyakorlatilag nyomógombokból lettek kialakítva. A jobb oldalra az oldal témáját tartalmazó szöveges rész. Az oldalak színvilága kapcsolódik az index.htm oldal színvilágához.



A lap baloldalán elhelyezkedő nyomógombok minden esetben egy linket tartalmaznak, amelyekkel elérhetővé válnak a portál további lapjai.

A portál menüterve és dialógus terve a következőképpen alakul:



1. szerencserol.htm
2. jatekszabalyok.htm
3. szimulator.htm
4. nyerni.htm
5. tortenetek.htm
6. blakjack.htm
7. automatak.htm
8. regisztracio.php

9. terkep.htm

Össességében elmondható, hogy minden egyes lap megnyitható a másik oldalról, rugalmas, könnyen átlátható szerkezetet biztosítva a felhasználó számára

Az oldalak létrehozása egyszerű másolással történt, így nem kellett csupán egyetlen sort megváltoztatni az aktuális honlap nyomógombok oszlopában és nem kellett mást tenni, mint az oldalakat egyszerűen fel tölteni tartalommal. Az oldalak forráskódjának közös része alapján elmondható, hogy tulajdonképpen egy tábla három különböző méretű cellájában helyezkedik el a tartalom. A felső részen a kép került elhelyésre. ``

A honlap bal oldali cellájában a nyomógombok pozíciója és mérete rögzített így pontosan egymás alá elhelyezhetők lettek, függetlenül attól hogy milyen hosszúságú szöveg került a value attribútumnak történt átadása során.

```
<div align="left">
  <table border="2" cellpadding="0" cellspacing="0" style="border-collapse: collapse"
bordercolor="#111111" width="643" height="532" id="AutoNumber1">
  <tr>
    <td width="643" height="75" colspan="2"></td>
  </tr>
  <tr>
    <td width="138" height="453" bgcolor="#993333">
      <input type="button" style="position:absolute; top:95;width:127;background-
color:#0066cc;cursor: pointer" value="Kezdőlap" name="B1"
onclick="document.location.href ='index.htm'">
      <input type="button" style="position:absolute; top:120;width:127;background-
color:#0066cc;cursor: pointer" value="A szerencséről" name="B1"
onclick="document.location.href ='szerencserol.htm'">
      <input type="button" style="position:absolute; top:145;width:127;background-
color:#0066cc;cursor: pointer" value="Játékszabályok" name="B2"
onclick="document.location.href ='jatekszabalyok.htm'" >
      <input type="button" style="position:absolute; top:170;width:127;background-
color:#0066cc;cursor: pointer" value="Veszélyek" name="B4"
onclick="document.location.href ='veszelyek.htm'">
```

```

        <input          type="button"          style="position:absolute;
top:195;width:127;background-color:#0066cc;cursor:  pointer"      value="Nyerni?"
name="B5" onclick="document.location.href ='nyerni.htm'">
        <input          type="button"          style="position:absolute;
top:220;width:127;background-color:#0066cc;cursor:  pointer"      value="Történetek"
name="B5" onclick="document.location.href ='tortenetek.htm'">
        <input          type="button"          style="position:absolute;
top:245;width:127;background-color:#0066cc;cursor:  pointer"      value="BlackJack"
name="B5" onclick="document.location.href ='blackjack.htm'">
        <input          type="button"          style="position:absolute;
top:270;width:127;background-color:#0066cc;cursor:  pointer"      value="Automaták"
name="B5" onclick="document.location.href ='automatak.htm'">
    </td>
    <td width="501" height="453" bgcolor="#ccff99">&nbsp;</td>
</tr>
</table>
</div>

```

Meg kell jegyezni azonban, hogy két oldal tartalmát tekintve eltér a fent említett oldalak struktúrájától. Az egyik a honlap-térkép amely egyszerű, csupán linkeket tartalmazó oldal, valamint a regisztrációs oldal, amely egy formon került kialakításra és beviteli mezőket és gombokat tartalmaz.

A rulett.php oldal

A rulett.php a portál legfontosabb eleme. A rulett népszerűsége miatt egyre inkább terjed az Interneten is. Azt nagyon fontos elmondani, hogy tisztán szerencsejáték. Lehetséges szisztémákat statisztikákat felépíteni, de a játék kimenetele eredeti kialakításánál fogva, abszolút a szerencsétől függ. Ez a tény teszi lehetővé, hogy véletlenszám-generátor segítségével előállított szám alapján tudjuk elkészíteni a játékot. Természetesen más, a lejtős körpályán érintőirányban, meghatározott kezdősebességgel indított golyó pályájának szimulációját megvalósító programot is lehet írni, de a mozgás differenciál egyenleteinek a leírása nem lenne tökéletes, hiszen minden fizikai körülményt nem lehet figyelembe venni.

A játék teljes egészében kliens oldali javascript nyelven íródott.

	1-12				13-24				25-36				
0	3	6	9	12	15	18	21	24	27	30	33	36	1-st
	2	5	8	11	14	17	20	23	26	29	32	35	2-nd
	1	4	7	10	13	16	19	22	25	28	31	34	3-rd
	low 1-18			even	red	black	odd	high 19-36					

Tőke	Tét	Nyeremény	Szám	Idő
200				8

Kilépés

A bejelentkezés után megjelenő kezelőfelület, tartalmazza az eredeti játék minden funkcióját. Az indítás gomb megnyomása után 20 másodperc idő áll rendelkezésére a játékosnak a tétek megtételére, természetesen a tőkéje erejéig.

A tétek megtétele a megfelelő számra, illetve helyre történő pozicionálással történik. Azon a helyen, ahol az egérmutató kézzé változik a bal egérgomb megnyomása után sárga négyzetben megjelenik a tét. Többszöri egérekattintásra a tét értéke növekedik. Lehetőség van három előre programozott kombináció megtételére is. A nagyszéria, kisszéria, orphan gombokra történő kattintáskor, a program meghatározott pozíciókban helyezi el a téteket. További funkció, az utolsó tét törlése.

Ha lejárt a tét megtételére szánt idő, a program véletlenszám-generátora 0 és 36 között képez egy számot, amely alapján kiszámolja a nyereményt és hozzáadja a tőkéhez. Ha elfogy a tőke, nincs lehetőség a további játékokra. Ha tovább szeretne játszani a játékos, ismételten be kell lépni a játékba.

A program felépítése egyszerű, áttekinthető struktúra.

A kezelő felület legfontosabb eleme a játékasztal, amely a képernyő 0,0 pozíciójába lett elhelyezve. A játékasztal nem más, mint egy bitmap állomány.

```

```

A kép pozíciója nagyon lényeges, mert ehhez a helyzethez képest hozza létre a program a téteket tartalmazó objektumokat. A pozíció nem módosítható, mert a létrehozott objektumok helyzete is rögzített.

Az oldal betöltődésekor az onload esemény bekövetkeztében meghívásra kerül a létrehozdivék nevű eljárás. Az alábbi kódrészlet azt mutatja, hogy húsz oszlopban és hat sorban hozza létre a div konténereket amelyeknek megadja a pozícióját ,a háttér színt láthatatlan értékre állítja, valamint a kurzor alakját pointer-re állítja be. Ezzel a művelettel azt érjük el, hogy ha az egér a meghatározott pozícióba kerül, megváltozik az egérmutató alakja ezzel jelezve, hogy erre a helyre tét rakása lehetséges. Ha ezen a helyen egér kattintás történik meghívásra kerül az egyik számítást elvégző számolegyszeres('egyszeres'+j+'') eljárás .

```
oszlop=20;  
sor=206;
```

```

for (j=0;j<6;j++)
{
    oszlop=oszlop+74;
    sd=sd+"<div id='egyszeres "+j+" ' onclick=szamolegyszeres('egyszeres"+j+"')
style='background-color:none;cursor:pointer;
position:absolute;top:"+(sor)+"px;left:"+(oszlop)+"px;height:15px;      width:12px;z-
index:8'> </div>";

    egyszeres[j]=0;
}

```

Ez a kódrészlet fogja létrehozni a div objektumot.

```
document.getElementById("sokdiv").innerHTML=sd
```

Az összes objektum létrehozása után el kell indítani a visszaszámlálást. A visszaszámlálást gyakorlatilag a számol eljárás fogja megvalósítani.

Az eljárás lelke az a rekurzív rutin amely önmagát hívja meg 1000 miliszekundumonként azaz másodpercenként.

```
setTimeout("számol()", 1000);
```

Az eljárásban be van építve egy védelem arra az esetre, ha valaki netalántán megnyomná a visszaszámlálás közben a számol gombot. Állapot jelző változók (áll, megy) alapján a rutin el tudja dönteni, hogy zajlik-e a visszaszámlálás művelete, vagy sem.

```

function indít()
{
    if (számolás=="áll")
    {

        alaphelyzet();
        idő=20;
        számol();
    }
}

```

```

function számol()
{
  if (idő==0)
  {
    számolás="áll";
    kiértékel();
    clearTimeout();

  }
  else
  {
    setTimeout("számol()", 1000);
    idő=idő-1;
    document.getElementById('idő').innerHTML=idő;
    számolás="megy";
  }
}

```

A meghatározott pozícióba elhelyezett téteket az alábbi kódrészlet alapján lehet megérteni:

```

function szamok(azon)
{

  l=divekszama[azon.substr(1,1)][azon.substr(2,2)];
  l++;

  if (l>tóke)
  {
    alert("önnek nincs elég tőkése a tét megtételéhez!");
  }
  else

```

```

{
tét=tét+1;
document.getElementById("tét").innerHTML=tét;
tőke=tőke-1;
document.getElementById("tők").innerHTML=tőke;
divekszama[azon.substr(1,1)][azon.substr(2,2)]=1;

```

```

document.getElementById(azon).innerHTML=divekszama[azon.substr(1,1)][azon.substr(2,2)];
document.getElementById(azon).style.backgroundColor="yellow";
l=0;

```

Az l változóba teszi bele a korábban elhelyezett tétet, majd ha erre a tétre kattintás történik egyel nő az l változó értéke. Nő a tét egyel, és csökken a tőke értéke egyel. A documentum.getelementbyid.innerHTML tulajdonság megjeleníti a tőke és a tét aktuális értékét, valamint az adott pozícióban kiírásra kerül sárga háttérrel a tét értéke.

Ha a tétek megtételére szánt idő letelt, meghívásra kerül a kiértékel eljárás
Amelynek legfontosabb sora a véletlen szám generátor függvény:

```
szam=Math.round(Math.random()*36);
```

A függvény által a szam változónak átadott értékét egy switch szerkezet segítségével, a generált számhoz tartozó rutint hajtja végre a program.

Az algoritmus gyenként kiszámítja, hogy mely tétet kell figyelembe venni és a megfelelő tétek szorzóval felszámított egységei elhelyezésre kerülnek a nyereménypont nevű változóban.

A végeredmény táblán történő megjelenítését az alábbi rutin végzi

```

tőke=tőke+nyereménypont;
document.getElementById('tők').innerHTML=tőke;
document.getElementById('egyenleg').innerHTML=nyereménypont;
document.getElementById('egyenleg').style.backgroundColor="red"

```

```
document.getElementById('nyertesszam').innerHTML =szam;
```

A különleges tétek megjelenítésére egyedi kiírató rutin szükséges, hiszen minden egyes meghatározott pozícióban meg kell jeleníteni a div objektumokat.

A játékban, lehetőség van az utolsó tétek törlésére is. Abban az esetben, ha a játékos vissza akarja vonni a megtett tétjét akkor azt minden további nélkül megteheti. Az utolsó tét törlése gomb onclick eseménye meghívja a töröl nevű eljárást. Az eljárás az aktuális pozíció alapján törli, azaz átírja a div objektum háttér stílusát átlátszó értékre (transparent) és a háttérszínét törli, valamint üres értéket ad az objektum innerhtml tulajdonságának.

```
document.getElementById(azon).innerHTML="";  
document.getElementById(azon).style.background='transparent';  
document.getElementById(azon).style.backgroundColor="none";
```

Ahhoz, hogy a törléssel visszakapjuk a beírás előtti állapotot az alábbi kód segítségével valósítottam meg.

```
tőke=tőke+egyszeres[azon.substr(9,1)];  
document.getElementById("tök").innerHTML=tőke;  
tét=tét-egyszeres[azon.substr(9,1)];  
document.getElementById("tét").innerHTML=tét;  
egyszeres[azon.substr(9,1)]=0;
```

A portál üzemeltetése, biztonsági kérdések

A portál nyilvános honlap, amelynek minden egysége szabadon olvasható. Adattartalmat nem lehet feltölteni a portálra. A rulett játékprogram elérése regisztrációhoz kötött. A regisztráció nem tartalmaz különösebb személyes adatokat csupán a felhasználó nevét, lakcímét és e-mail címét kéri, amelyek általában is nyilvános adatok. A felhasználói név megadása az egyetlen adat, amely kötelező. A felhasználói név tulajdonság az adattáblában azonosítóként szerepel, amely alapján a felhasználó egyértelműen meghatározható. A rulett játékba történő belépéshez elengedhetetlenül fontos, hogy a felhasználói név és a jelszó szerepeljen az adatbázisban. A fenti két adat nélkül nem lehetséges a belépés. A felhasználó miután kitöltötte a kezdőoldalon található űrlapot az adatok elküldésére kerülnek az ellenőriz.php állomány számára. A vizsgálat során egy session indul, amely csak akkor fut ha már a sikeres belépés megtörtént értéke 1 lesz.

```
session_start();
    $_SESSION["auth"]=1;
    header("Location: rulett.php");
    exit;
}
else
{
    header("location: index.htm");
    exit;
}
```

A rulett.php állomány elején található egy rövid php scrip amely megvizsgálja a session állapotát

```
<?php
session_start();
if(!isset($_SESSION["auth"])) {
header ("location: index.htm");
```

exit;
}

Ha session nem fut, azaz értéke nem 1, abban az esetben nem lehetséges a rulett kódjának elérése és visszadobja a felhasználót az index állományra, hogy a belépési, vagy regisztrációs eljárást végrehajtsa.

Tehát, abban az esetben, ha egy illetéktelen felhasználó a böngészőben közvetlenül próbálja elérni a rulett játékot, nem tudja megtenni, mert a program automatikusan visszadobja az index.htm oldalra.

A portál üzemeltetése.

Amikor honlapot szeretnénk üzemeltetni érdemes megvizsgálni annak gazdasági hátterét és eldönteni, hogy a vállalkozás szempontjából mely megoldások a legkifizetődőbbek:

	Internet-szolgáltató	Saját szerver	Szerver bérlet
Felállítás költségei (Ft):			
Domain regisztráció	5.000-8.000	5.000-8.000	5.000-8.000
Honlap tervezés	70.000-150.000	70.000-150.000	70.000-150.000
Hardver		150.000-200.000	
Szoftver		20.000-500.000	
<i>összesen:</i>	<i>75.000-158.000</i>	<i>245.000-858.000</i>	<i>75.000-158.000</i>
Üzemeltetés költségei (Ft/hó):			
Domain karbantartás	1.000-2.000	1.000-2.000	1.000-2.000
Tárhely bérleti díj	5.000-20.000		
Szerverkapcsolat		28.000-70.000	
Szerver bérlet			65.000-85.000
Hardver karbantartás		5.000-10.000	
<i>összesen:</i>	<i>6.000-22.000</i>	<i>34.000-82.000</i>	<i>66.000-87.000</i>

A táblázat alapján elmondható, hogyha internetes vállalkozásban gondolkodunk, vagy csupán csak szeretnénk, ha a vállalkozásunk bemutatása kikerüljön az Internetre annak meglehetősen költségei vannak.

Jelen esetben egy „amatőr” honlap elkészítéséről van szó, amelynek költségei csupán a domain regisztrációra terjednek ki.

A rendszer üzemeltetése

Jelen esetben a portál egy otthoni kiszolgálói környezetbe lett telepítve. A konfiguráció védelméről kétszintű tűzfalrendszer gondoskodik. A rendszer első védelmi vonalát hardveres tűzfal biztosítja intelligens csomagszűréssel (stateful packet inspection) . Meg kell jegyezni, hogy az intelligens csomagszűrés elegendő az internetes betörések megakadályozására, tehát ebből a szempontból a rendszer túlméretezett, robusztus.

A második védelmi vonal a szerver számítógépre telepített rezidens tűzfal szoftver, rezidens víruskereső és spyware killer program.

A honlap üzemeltetéséhez el kell helyezni a portál állományait a webservert azon könyvtárban, amely elérhető a felhasználók számára. Ez a könyvtár lehet az appserv alkalmazás esetén az appserv/www könyvtára. Nagyon lényeges a megfelelő, rendszeres naplózás és annak helyes értelmezése kiértékelése.

Összefoglalás

A szakdolgozat első részében tárgyalásra kerültek a html, javascript, php nyelvek amelyek megalapozzák a további részekben bemutatott honlap terv és annak megvalósítása elméleti hátterét. További részekben néhány szóban említésre került a két legelterjedtebb webservert alkalmazás az Apache és az IIS. A szakdolgozat központi eleme a rulett szimulátort tartalmazó szerencsejátékokról szóló portál, amelyben felhasználásra kerültek a korábban tárgyalt elméleti ismeretek. A portál bejelentkező képernyője és központi lapja az index.htm, amelyről a felhasználó tovább navigálhat a tartalmakhoz. A központi lapról léphetünk tovább a rulett szimulátorba. A szimulátor gyakorlatilag egy teljes mértékben javascriptben készített program, amely szinte lemásolja az elektronikus játékkeretekben használt gépet, funkciói tekintetében. A szimulátor eléréséhez használt beléptetés, lényeges része a portálnak, mivel csak regisztráció után lehet bejutni a szimulátorba. A beléptetés és regisztráció kihasználja a php nyelv lehetőségeit, és egyfajta biztonsági elemként munkafolyamatok alkalmazásával megvalósítható a jogosultság ellenőrzés. A beléptetés folyamata adatbázis kezelési feladat, amelynél a php-mysql szerver szimbiózisát használjuk ki a felhasználók lekérdezésénél, valamint a regisztrációnál. A portál megvalósításánál, keverednek a HTML CSS Javascript php mysql elemek. A portál esztétikája egyszerű, színvilága reményeim szerint nem túl hivalkodó. Elsősorban a portál funkcióinak maradéktalan megvalósítása volt a cél. A rendszer egyik gyenge pontja a javascript szimulátora, amely konkrétan a Mozilla úgynevezett alternatív böngésző nyelvezetére íródott. Fontos továbbfejlesztési feladat az Internet Explorer-re történő adaptálás is. Ez azonban nagyon összetett feladat a teljes programstruktúra újragondolását igényli. A rendszer továbbfejlesztésének folyamatában a regisztráció, illetve a felhasználó követés kifinomultabbá tétele a feladat. Jelenleg nagyon egyszerű struktúrát alkalmazok, nem elégséges, a pontos adminisztrációs feladatok megvalósításához. Jelen pillanatban a honlap amatőr nonprofit rendszerben működik nem igényli összetettebb rendszer megvalósítását.

A program fejlesztésekor fellépő nehézségek abban jelentkeztek, hogy a javascript nyelvnek nincs integrált fejlesztői környezete. Igyekeztem a kódot áttekinthető modulok összességeként megvalósítani, megjegyzésekkel ellátni, de így is a hibakeresés megoldása nagy figyelmet követelt. A portál elkészítése során következik a rendszer

üzemeltetése. Először az appserv alkalmazás került feltelepítésre. A szervert biztonsági szempontból védelemmel kellett ellátni, mert ha fejlesztési időben portscanneléses technikával rátalálnak a rendszerre, akkor betörési kísérlet alkalmával nagy károkat okoznának a rendszerben. A statisztika azt mutatja, átlagosan napi 10-20 ilyen kísérlet történt a rendszer fejlesztésének idején. Az illetéktelen behatolási kísérletek kivédésére a webservert alkalmazás helyes konfigurálása harveres tűzfal és a rezidens szoftveres tűzfal, vírus és egyéb védelmi programok használata jelenti a megfelelő védelmet.

Köszönetnyilvánítás

Szeretnék köszönetet mondani családomnak akik türelemmel viselték estéimen, azt hogy nem velük törődtem, hanem célom elérése érdekében éjszakákba nyúlóan tanulmányaimra és szakdolgozatom elkészítésére koncentráltam.

Szeretném megköszönni tanárainknak, kollégáimnak azt a sok szakmai segítséget amely nélkül nem jött volna létre ez a dolgozat és a portál. Nem utolsó sorban köszönetet mondok Füvesi Gábornak, aki gyakorlati tapasztalatával, bevezetett a szerencsejátékok világába és óvott attól, hogy bármilyen káros hatással legyen rám ez a szórakozás.

Miskolc, 2007. december 10.

Török Attila

Informatika Tanár jelölt

Felhasznált irodalom

Julie C. Meloni **Kiadó:** [Kiskapu Kft.](#) 2003 **ISBN:** 9639301493

Gál Tibor **Kiadó:** [Műegyetemi Kiadó](#) 2005

Holczer József **Kiadó:** [Jedlik Oktatási Stúdió Bt.](#) 2005 **ISBN:** 9638651490

Matt Zandstra **Kiadó:** [Kiskapu Kft.](#) 2004 **ISBN:** 9639301868

[Bóc István](#) A szerencse titkai Kiadó:Typotex Kft. Elektronikus Kiadó - 2004
ISBN:9639548286

BÓC István **Gondolkodom, tehát nyerek -**
Taktikák és praktikák: lottó, kaszinó, tévévetékedők, tőzsde ISBN: 963 9132 86 1

Internetes portálok:

<http://www.w3schools.com/default.asp>

<http://htmlspec.web.elte.hu/>

<http://htmlgoodies.earthweb.com/primers/jsp/>

<http://www.honlaptervek.hu/honlaptervek/index.php>