

**SZEGEDI TUDOMÁNYEGYETEM
BÖLCSESZETTUDOMÁNYI KAR
SZEGED**

EGY TÖRTÉNELMI FORRÁS HTML ALAPÚ FELDOLGOZÁSI LEHETŐSÉGE

**LÓNYAY MENYHÉRT POLITIKAI NAPLÓJA
(1861. JANUÁR 1. - 1861. MÁJUS 10.)**

**Készítette: Berkes Hajnalka
IV. évfolyam könyvtári informatika**

**Témavezető: Sándor Ákos
rendszergazda
SZTE-EK**

2003.

TARTALOMJEGYZÉK

I. Bevezetés

II. Előkészületek: az alapanyag előállítása

III. Megvalósítás: a CD-ROM készítésének menete

IV. Formatervezés: a CD-ROM formázást definiáló elemei

V. Tesztelés

VI. Összegzés

IRODALOMJEGYZÉK

MELLÉKLETEK

I. Bevezetés

Szakdolgozatomban egy Microsoft Word szövegszerkesztővel készített dokumentum HTML alapú feldolgozását, az átalakítás során felmerült problémák ismertetését, a munkafolyamat rögzítését tűztem ki célul.

Témájaként egy történelmi forrást, Lónyay Menyhért 1861. január 1. és 1861. május 10. közötti naplóját választottam, mivel történelem diplomamunkám is erről szólt, és meg akartam valósítani a napló HTML alapú feldolgozását.

A CD-ROM a neoabszolutizmus végének értékes forrását tartalmazza, mivel a Lónyay-napló egy, a korszakban döntő jelentőségű politikai eseményt, az 1861-es országgyűlést és annak közvetlen előzményeit tárgyalja. Szerzője a vezető politikai körökhöz tartozva precízen számol be az 1848 óta nem lévő eseményekről (követválasztás, megyei élet felporzódása, Ausztriával való kapcsolat rendezésének lehetősége stb...), a korabeli politikai élet szereplőiről, azok álláspontjairól. Naplójellegű forrás, azaz az objektív beszámolókon túl szubjektív bejegyzésekkel, vélemény-nyilvánításokkal is találkozhatunk benne. A feldolgozást, mivel számtalan eseményt, személyt aprólékos részletességgel, alaposággal említ, hasznosan forgathatja az érdeklődő és kiindulási alapként szolgálhat a kutató számára is.

Az offline weblap (CD-ROM) felépítése során a tartalom és a megjelenés technikai és tartalmi eszközeinek harmonizálására törekedtem, a HTML hierarchiát úgy próbáltam kialakítani, hogy abban a Word dokumentum szerkezete visszatűnjék.

Mindent összevetve egy dinamikus, esztétikus, könnyen áttekinthető, egyszerűen használható CD-ROM kialakítása volt a célom, amely éppúgy használható oktatási segédanyagként, mint kutatási segédletként.

Dolgozatom hátralévő részében a készítés során felmerülő problémákat, a megvalósulási lehetőségeket valamint a felhasznált HTML és Javascript elemeket mutatom be.

Mellékleteként hozzacsatolom a technikailag érdekesebb böngészőoldalak kinyomtatott változatait, valamint a kész CD-ROM-ot. Utóbbi beállításainak optimális megtekintéséhez Microsoft Internet Explorer 4.X-es böngésző használata és 1024*768-as képernyőfelbontás ajánlott!

II. Előkészületek: az alapanyag előállítása

A CD-ROM konkrét elkészítésének megkezdése előtt két feladattal kellett megbirkóznom. Egyrészt a már meglévő Word dokumentumot HTML alapúvá kellett alakítani, másrészt a szöveganyagot további anyagokkal (pl. képekkel) kellett bővíteni.

Az írott szöveget először Wordben próbáltam HTML formátumba elmenteni. Az átkonverálás során a következő problémák adódtak:

1. A Word generálta HTML kód meglehetősen sok felesleges információt, HTML tag-et tartalmazott,

2. A HTML-es formátumban nem jelentek meg az eredeti szöveg lábjegyzetei.

Utóbbi nehézség megoldása azért bírt kiemelkedő jelentőséggel, mert a Word szöveg - forrásfeldolgozás lévén - körülbelül felét lábjegyzetek tették ki. Ezek Wordben meglévő formátumát (a lap alján folyamatosan növekvő sorszámozásban) azonban a HTML kód nem tükrözte, sőt az átalakítás során a lábjegyzeteket teljesen figyelmen kívül hagyta. További feldolgozásukra csak fejezetenként külön HTML file-okba való elmentés után gondolhattam.

Az első probléma megoldását az általam használt HTML szerkesztő program, a Dreamweaver UltraDev 4.0-ás változata kínálta a „Clean Up Word HTML” funkciójával. Segítségével beállíthatóak azok a HTML tag-ek, amelyeket el szeretnénk távolítani a Word generálta HTML forráskódból, így tisztítva azt meg a felesleges elemektől.

A művelet elvégzését követően a szöveg még formázatlan volt, formázását a továbbiakban stíluslapok segítségével valósítottam meg.

Az anyaggyűjtés egyik sarokkövét jelentette a naplójegyzetekhez kapcsolódó korabeli képanyag felkutatása. Mivel a napló 1861-ben íródott és a fotográfia ekkor még meglehetősen kezdeti szakaszát, gyermekkorát élte, nem csupán fényképeket, hanem a korabeli újságok (pl. Vasárnapi Újság, Hazánk s a külföld...) rézkarcolatait is felhasználtam a napló szövegének illusztrálására.

A képeket változatos forrásokból (korabeli újságok képanyaga, Internet, képes albumok, CD-ROM-ok...) próbáltam gyűjteni, a lehetőségekhez mérten a minél pontosabb, teljesebb anyaggyűjtésre törekedtem.

A képek egy része digitalizált formában jutott hozzám (Internetről, CD-ROM-okról), másik részét magam szkenneltem Hewlett-Packard PrecisionScan Pro szoftver segítségével 150-300 dpi felbontásban, 256 színárnyalatú színes képként. A szkennelés során a képeket képaláírásukkal együtt szkenneltem be, de a későbbi munka során a képaláírásokat az oldalakba való beillesztésükkor nem használtam fel. Az egységes formatervezés érdekében a továbbiakban minden egyes képhez külön aláírást készítettem. A szerzői jogokat szem előtt tartva tájékoztató jelleggel a képek forrását is feltüntettem alattuk. (Ez az Internetes források esetében azt jelenti, hogy azok nem linkként funkcionálnak!)

Az összegyűjtött képek feldolgozást (retusálását, méretezését, egységes színezését...) a Jasc Software Paint Shop Pro 7.02-es programjával végeztem el.

III. Megvalósítás: a CD-ROM készítésének menete

A.) A használt programok

A CD-ROM elkészítésének megkezdése előtt megismerkedtem a használni kívánt programokkal. A weblap összeállításához alapvetően két programot alkalmaztam: a Macromedia cég által készített Dreamweaver UltraDev 4 nevű HTML szerkesztő szoftvert és a Xara cég Webstyle 2.0-ás webgrafika készítő programját. Lényegében az egész weblap szerkesztését a Dreamweaver segítségével oldottam meg. Próbáltam a program lehetőségeit a CD-ROM-hoz idomulva maximálisan kihasználni. Ezeket bővítettem a Webstyle technikai adottságaival.

B.) Az index.htm(l) oldal

A konkrét megvalósítás első fázisának a HTML hierarchia megalkotása számított. A struktúra kialakításánál tartalmi szempontokat és az eredeti Word dokumentum felépítését vettem figyelembe.

Az indítóoldal (ld. 1. számú melléklet) az index.htm illetve az index.html nevet kapta, mivel bevett gyakorlatnak, ajánlott indítóoldalnak számít ez a file-név a Világhálón. Az oldalon Lónyay Menyhért 1860-61-es naplójának képe látható, amelyre való klikkelés a CD-ROM-on való továbblépést teszi lehetővé.

A dinamikusságot a **Javascript alkalmazások** viszik az oldalba. Ezek a scriptek tulajdonképpen kis programok, amelyek a HTML oldallal együtt vagy azokba ágyazva töltődnek le. A forráskódban itt a HTML oldalak fejléceibe találhatók a <script></script> címkék között. A böngészők ezen alkalmazásokat még letöltés közben, először, soronként hajtják végre. Ha valamelyik browser nem tudná értelmezni őket a <!-- és //--> karaktereknek köszönhetően átugorja a parancssort. Nem kell hozzájuk értelmező részt készíteni, a böngésző automatikusan értelmezi és tölti be őket (interpreter jelleg). A program a következő szerkezeti egységekből épül fel:

1. utasítások;
2. beépített objektumok, azok tulajdonságai, függvényei, eseménykezelői,
3. a felhasználó által definiált függvények, változók, objektumok.

Javascript állítja be, pl. a *linkek színét*. (Mivel a háttérszín ekképpen való definiálását nem mindegyik böngésző - pl. Mozilla 1.0, Opera 6.01 - értelmezte, a háttérszínt stíluslapon is megadtam!) A beállítás így néz ki a forráskódban:

```
<SCRIPT language="JavaScript">
<!--
document.vlinkColor="775b4d"
document.linkColor="c5c2a5"
document.alinkColor="CCCC99"
//-->
</SCRIPT>
```

A *státusbárban futó tájékoztató szöveg* szintén Javascript alkalmazás, programját magam állítottam össze. Ennek forrása:

```

<SCRIPT language="JavaScript">
<!--
var text = "A belépéshez klikkeljen a naplóra!";
var length = text.length;
var width = 100;
var pos = 1 - width;
function scroll()
{
    pos++;
    var scrolled = "";
    if (pos == length) pos = 1 - width;
        if (pos < 0)
        {
            for (var i = 1; i <= Math.abs(pos); i++)
                scrolled = scrolled + " ";
            scrolled = scrolled + text.substring(pos, width -i +1);
        }
        else
        {
            scrolled = scrolled + text.substring(pos, width + pos);
        }
    window.status = scrolled;
    setTimeout("scroll()",250);
}
//-->
</SCRIPT>

```

A program működésének alapja négy globális változó - „text”, „length”, „width”, „pos” -, melyeknek számértéke a futtatás során folyamatosan változik, ez okozza a státussorban lévő szöveg „folyását”. A „text” változó a státussorban kiírandó szöveget tartalmazza, a „length” ennek karakterben mért hosszát, a „width” azt a számértéket, amilyen szélességet igénybe vesz a szöveg a státussorban való futás során, a „pos” a folyás „mértékét”. (Azaz, hogy előző meghívása óta hány karaktert lép előre a szöveg!) Az utolsó három változó induló értéke sorrendben a következő: 35, 100, -99.

Az értékek változását a „scroll()” paraméter nélküli függvény deklarálásával értem el.

Mit is tartalmaz ez a függvény? Benne a pozíciót („pos”) rendre növeljük eggyel és definiálunk egy „scrolled” nevű globális változót, melynek kezdőértéke egy szóköz. Ezt két feltételes elágazás követi; ha a „pos” változó értéke egyenlő a szöveg hosszának értékével, előbbi a következő értéket veszi fel: „pos = 1 - width”. Abban az esetben viszont, ha a pozíció kisebb nullánál lefut az alábbi számlálós ciklus:

```

for (var i = 1; i <= Math.abs(pos); i++)
    scrolled = scrolled + " ";
scrolled = scrolled + text.substring(pos, width -i +1);

```

A vezérlési szerkezet addig hajtódik végre, amíg a „pos” változó abszolút értéke nagyobb egyenlő, mint „i” értéke. Ha ez igaz, akkor a „scrolled” változó értékét konkatenációval szóköz plusz a „text” változóból a következő módon kivágott szövegrésze állítjuk: „text.substring(pos, width -i +1)”; ha hamis a „scrolled” értéke ekképpen alakul: „scrolled = scrolled + text.substring(pos, width + pos)”.

Ezek után már csak ki kellett íratatni és megfelelő időközönként meg kellett hívni a „scrolled” változó aktuális értékeit. A státussorban történő kiíratást a Window objektum status elemtulajdonságával értem el. Utóbbi feladatot a Date objektum setTimeout nevű beépített függvénye segítségével oldottam meg. A függvény két paramétert tartalmaz: első

paraméterében a „scroll()” függvényt, amit a második paraméterében meghatározott 250 darab ezred-másodpercenként (azaz negyed-másodpercenként) hívja meg.

Az **üdvözlőablakot**, amely a CD-ROM elindítását követően töltődik be és annak működési feltételeiről, optimális futtatási környezetéről ad információkat a következő Javascript program futtatja:

```
<BODY onLoad="alert( 'Üdvözlöm! A CD-ROM beállításainak optimális megtekintéséhez Microsoft Internet Explorer 4.0-ás böngésző használata és 1024*768-as képernyőfelbontás ajánlott! Amennyiben nem tudja milyen böngészőt használ a nyitóoldalon klikkeljen a Böngésző típusa nevű linkre!' )" >
```

A felbukkanó ablakot a Window objektum alert nevű beépített függvénye adja, amely egy OK gombos párbeszédablakban kiírja a paramétereiben megadott, string-típusú üdvözlőszöveget. Az ablak indítás utáni betöltéséért az onLoad eseménykezelő felel, amelyet a Body egyik paramétereként definiáltam.

A nyitóoldalon található **„Böngésző típusa” nevű link** a felhasználó browseréről ad információkat. Segítségével megtudhatja böngészőjének típusát és verziószámát, annak alapbeállításainak megfelelően. (Emiatt nevezi például a Mozilla 1.0 és a Netscape 6.1-es Netscape 5-nek magát!) Az alkalmazást az alábbi Javascript kód állítja elő:

```
<A HREF="#" onClick="MM_callJS('alert ( \'&quot;Ön &quot; + navigator.appName +\&quot; &quot; + parseFloat(navigator.appVersion) +\&quot; típusú &quot;+\&quot; böngészőt&quot; + &quot; használ!\&quot; )')") > Böngésző típusa</A>
```

A kódot a Dreamweaver beépített Javascript hívó opciójával készítettem, ami egy előzőleg definiált MM_callJS függvény paramétereként hívja meg a programot.

A program alapja a Window objektum alert nevű beépített függvénye, amely egy OK gombos párbeszédablakban kiírja a paramétereiben megadott értéket. Az érték jelen esetben egymással - + jellel - összefűzött stringekből (szöveges karakterláncokból) és a Browser objektum két elemtulajdonságának (appName, appVersion) lekérdezéséből tevődik össze. A böngésző verziószámának kiírásához a parseFloat nevű beépített függvényt használtam, amely kimenete egy valós számérték. Utóbbi alkalmazás a legszebben az Opera 6.01 böngészőben látható: „Ön Opera 6.01 típusú böngészőt használ!”

Az alkalmazás természetesen - a már fentebb leírt üdvözlőablak szerkezete alapján -, a Dreamweaver beépített funkciója nélkül is megoldható; jelen eset a technikai sokféleséget igyekszik reprezentálni.

IV. Formatervezés: a CD-ROM formázást definiáló elemei

Az indítóoldal elkészítését a HTML oldalak kialakítása, formázása követte. A formatervezés során egységes szerkezetre, harmonikus szín- és formavilágra törekedtem figyelembe véve a tartalom igényeit. Így esett a választásom a különböző pasztellszínekre, a hozzáillő háttérre, betűtípusra illetve a képek látható színvilágára. Mind a színek, mind a háttérkép kiválasztását a Webstyle program segítségével végeztem.

A.) Stíluslapok

A formázóutasításokat az egységes megjelenítés miatt stíluslapok segítségével oldottam meg. A stíluslapoknak mindhárom típusát alkalmaztam a formázás során, azaz

1. az ún. belső stíluslapokat
2. az ún. külső stíluslapokat
3. és a `<style></style>` paraméterek közé zárt, a HTML forráskód fejlécében (head) elhelyezett formázóutasításokat.

1. Belső stíluslapok

Belső stíluslapokkal alakítottam ki az Időszalag menüpont alatt található Jump Menü háttérszínét, betűtípusát és betűszínét. Ennek forráskódja így néz ki:

```
<SELECT style = "background-color:#E7E2B8; font-weight:bold; font-  
style:verdana" ... >  
  <OPTION style = "color:#800000" ... >Kérem, válasszon!</OPTION>  
  <OPTION style = "color:#2D251E" ... >1861. január 3.</OPTION> ...
```

2. Külső stíluslapok

A külső stíluslapok készítését a Dreamweaver beépített alkalmazása segítségével oldottam meg. Ezek a HTML forráskódban a következőképpen jelennek meg:

```
<LINK REL="stylesheet" HREF="Szoveg/Alapformazas.css" TYPE="TEXT/CSS">
```

A szöveg alapformázását (alapértelmezett betűtípus, betűméret, betűszín, háttérkép, bekezdések, margók, szöveg elhelyezése stb...) az Alapformazas.css nevű file tartalmazza.

Minden, a szöveg alapformázásától eltérő formázóutasításra külön-külön külső stíluslapot definiáltam, amelyeket a hierarchiában az alkalmazott oldalakkal azonos szinten helyeztem el. Így például velük oldottam meg a naplóban található szövegkritikai megjegyzések ([...], [!], [?] karakterek), Lónyay Menyhért személyes kiemeléseinek, az idézeteknek, a füljegyzeteknek, a datálásoknak, a lábjegyzeteknek, a forrásfeltüntetéseknek a formázását is. Lsd. Kommentarkarakterek.css, Kiemelések.css, Idezet.css, Datum.css, Lábjegyzet.css, Forras.css stb...

3. <style> </style> között elhelyezett formázóutasítások

A <style> </style> között elhelyezett formázóutasítások között szerepel az oldalakon található linkek színeinek (link, visited link, active link) meghatározása:

```
<STYLE>
A{color:#c5c2a5}
A:visited{color: #191710}
A: active{color:#8c8162}
</STYLE>
```

B.) Frame-ek

A HTML oldalak kialakításának lényeges elemei a frame-ek (keretek). Mindvégig frame-es szerkezet adja a képernyő felosztásának alapját, azaz a nyitóoldalt követően bejövő oldal (ld. 2. számú melléklet) keretes szerkezete a későbbi használat során is megmarad. A három frame közül a bal oldali a navigációs bárt és a fölött elhelyezett analóg órát tartalmazza, a jobb oldalon felül lévő a CD-ROM címét, az alsó frame-ben indításkor a kinyitott napló képe található, de tartalma a használat során a menük tartalmától függően változik, azaz ebbe a keretbe töltődik be azok tartalma. A frame gyakran magába ágyaz újabb keretes szerkezeteket. Például „A napló politikai háttere” menübe való belépéssel újabb frame-es szerkezet töltődik be. (ld. 3. számú melléklet) Az egyes HTML oldalak adott keretbe való betöltését a target tag használatával, értékadásával értem el. A target tagnek fix értékeket (self, parent ...) és az elnevezett frame-ek neveit is adhatjuk értékül.

A frame-ek mindvégig „láthatatlanok” a felhasználó számára, azaz keret nélküliek.

A keretes szerkezet úgy hiszem legsikeresebb megvalósítását értem el *a Fogalomtár, a Névmutató és a Képtár menü szerkezetének* (ld. 8. számú melléklet) kiépítése során. Az egyes menükbe való belépést követően olyan frame-es szerkezet töltődik be, amely felső keretében az ABC betűi láthatóak, míg alsó keretében a menühöz tartozó fogalmak vagy nevek vagy, képcímek betűrendben. Az alsó keret hivatkozásai - a gördítősáv segítségével előrehaladva - önállóan is használhatóak, de a lista alkalmazása meggyorsítható az ABC egyes betűire való egérgattintással, amely az alfabetikus listát az adott betűnél nyitja meg. (Elkészítését az anchor nevű címkének és a target tagnek a használatával értem el.) A megnyitott linkek az alsó keretbe töltődnek be, így az ABC továbbra is látható és használatával lehetőség nyílik a listában való nagyobb ugrásokra is.

C.) A képek

A CD-ROM készítése során kiemelkedő szerepet tulajdonítottam a képek formázásának, hiszen nagyban hozzájárulnak az oldalak színesítéséhez. Ám az egyik legnagyobb problémát is a képek retusálása jelentette. A leggyengébb minőségűek (homályosak, sötétek, pixelesek...) a Magyar Életrajzi Lexikon CD-ROM változatáról letöltött képek voltak.

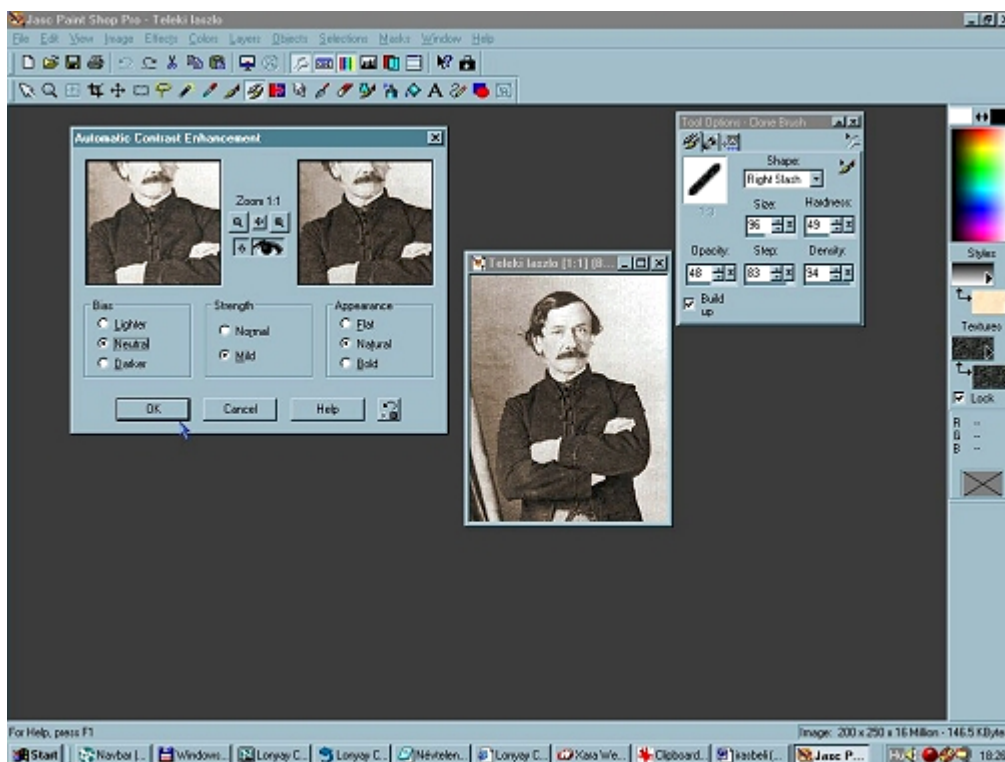
A retusálást a Paint Shop Pro 7.02-es software-rel végeztem. Alapvetően hármassal célkitűzéssel álltam neki a képek formázásának:

1. a képek minőségét javítani,
2. színvilágukat „régies” hangulatúvá tenni,
3. megfelelő méretűvé alakítani őket.

A képminőség javítására az eredeti képtől függően alkalmaztam a program beállításait: ahol kellett élesítettem, világosítottam, sötétítettem stb...

A színvilág kiválasztásánál régi, megbarnult fényképek készítésére törekedtem; s ahol nem feltűnően szép, színes fotókra (például Széchenyi István arcképe) bukkantam ezt a Paint Shop Pro Colorize illetve Automatic Contrast Enhancement opciójával valósítottam meg (lsd 1. számú képet). Némi nehézséggel a grey scale képekkel kapcsolatban találkoztam, ugyanis a kép betöltését követően nem élt a Colorize funkció. Így ezeknek a képeknek a színmélységét először az Increase Color Depth→16 Million Colors funkcióval megnöveltem, majd ezután alkalmaztam a fenti parancsokat (lsd. 1. számú képet).

A megfelelő méret kialakításának szükségességét a CD-ROM működésének az a sajátossága magyarázza, hogy azt szerettem volna benne megvalósítani, hogy eredetileg egy 120*150 pixel nagyságú képre, ha ráklikkelünk egy új HTML oldalon annak kinagyított változata töltődjön be. Ennek oka nem más, mint hogy a naplóban előforduló személynevekhez a gyorsabb betöltődés miatt olyan lábjegyzetformátumot szerkesztettem, amely bal oldalán látható a személy 120*150 pixel nagyságú képe, mellette pedig életrajz adatai. A képek kis- illetve nagy méretének kialakítását a Paint Shop Pro „Crop” opciójának használatával értem el.

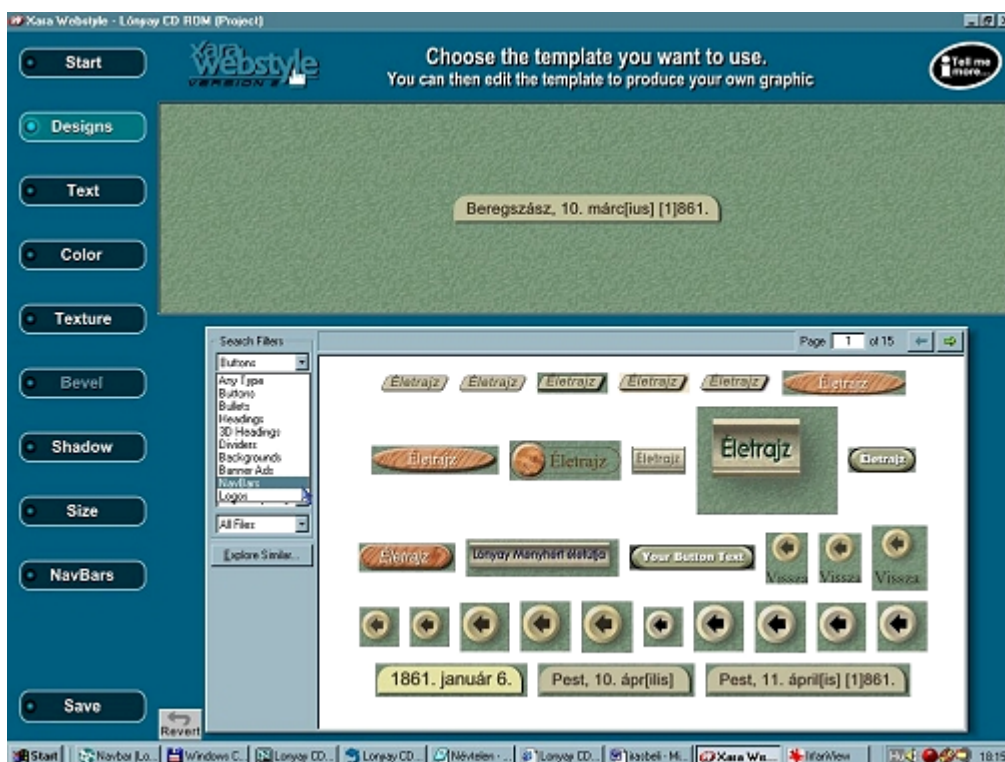


1. számú kép Paint Shop Pro munkaoldal: a képszín beállítása.

A fényképek HTML oldalba való helyezésekor az tag legtöbb paraméterét próbáltam kihasználni: pl. a képeket kerettel (border), alternatív szöveggel (alt) láttam el. Az országgyűlési képviselőket tartalmazó fénykép például Image mapként funkcionál, azaz a személyekre klikkelve azok életrajzaihoz juthatunk el. Megvalósítását szintén a Dreamweaver beépített funkciójának köszönheti.

Képként kerültek az oldalakba a menüpontokra klikkelve bejövő frame-es szerkezetekben a fejléccímek, a naplórészlet datálásai, füljegyzetei. Formázásukat (méretüket, színösszeállításukat) a Webstyle programmal készítettem el (lsd. 2. számú képet).

Ugyanitt állítottam össze a rollover image-ként működő nyomógombok (ld. 3. számú melléklet) alapanyagait is. A legördülő, hátsó képeket az eredeti elsőhöz képest élesebb kontrasztszínekkel láttam el, méretüket 1-2 pixellel kisebbre vettem. A képek rollover technikáját a Dreamweaver beépített alkalmazásával valósítottam meg.



2. számú kép Webstyle munkaoldal: a napló datálásainak elkészítése.

D.) Javascript alkalmazások

A Javascript programot a CD-ROM elkészítése során - nyitóoldalon említett alkalmazásokon kívül - számos helyen használtam. Javascript program vezérli a navigációs bár képcseréjét, a fölötté látható analóg órát, a státuszsorban történő kiíratásokat (ld. 2. számú melléklet), a „Vissza” link, a „Kilépés” menü (ld. 11. számú melléklet), a Jump Menü (ld. 9. számú melléklet) működését. Utóbbi egy legördülő menüben felkínálja a napló adott napon történő megnyitását.

Ezek egy részét a Dreamweaver beépített opciói segítségével készítettem el (navigációs bár képcseréje, Jump Menü működése, státuszbarba történő kiíratás), másik részét magam állítottam össze.

A saját magam által elkészített Javascript programok közé tartozik a „**Vissza**” link, amelyet helyzetétől, előfordulásától függően kétféleképpen definiáltam. Azokon az oldalakon, ahol az egy szinttel feljebb való visszalépést kellett megvalósítani (ld. 4., 5., 6. számú melléklet) forráskódja így néz ki:

```
<A HREF="javascript: history.back()">Vissza</A>
```

Ezt alkalmaztam például az egyes lábjegyzetekről való visszalépéskor, ami azért praktikus, mert a naplóban többször előforduló hivatkozások bármelyikéről ugyanoda ugrik vissza a böngésző, ahonnan kiindult.

A „Vissza” link másik értelmezésével azokon az oldalakon találkozhatunk, ahol több kép egymás után történő nézegetése lehetséges (ld. 7. számú lábjegyzetet). Az utolsó kép vissza linkje ebben az esetben - ha a képsor öt darabból áll - a következő forráskód szerint működik:

```
<A HREF="javascript: history.go(-5)" >Vissza</A>
```

Magam készítettem a **Kilépés menüt** vezérlő Javascript programot is, amely a Confirm nevű beépített függvény segítségével egy OK és egy Mégse gombot tartalmazó párbeszédablakban kiírja a paraméterében megadott értéket, jelen esetben egy a kilépés megerősítésére vonatkozó stringet („Biztos ki akar lépni?”). Amennyiben a felhasználó az OK gombot nyomja meg, a program bezárja a böngészőablakot (Microsoft Internet Explorer esetében egy újabb megerősítést követően!), Mégse gomb lenyomása esetén pedig a főoldalra ugrik. Az alkalmazás forráskódja ekképpen néz ki:

```
<SCRIPT language="JavaScript">
<!--
function kilepes()
{var logikai = confirm("Biztos ki akar lépni?" );
  if (logikai)
    { window.parent.close(); }
}
//-->
</SCRIPT>
```

Azaz definiáltam egy „kilepes” nevű függvényt, amelyben a logikai nevű lokális változó tartalmazza magát a confirm függvényt. A változó logikai értéket („true”, „false”) ad vissza. Amennyiben „true” értéket eredményez, azaz létezik a logikai változó (a felhasználó az OK gombot nyomta meg) a program bezárja az aktuális ablak szülőablakát, jelen esetben a frame-es szerkezet egészét, „false” értéknél pedig a főoldalra ugrik.

A program működéséhez a Kilépés linknél az onClick eseménykezelővel meg kellett hívni a „kilepes” nevű függvényt, a „href” paramétert a megfelelő értékre kellett állítani:

```
<A HREF="../../../Lonyay_cd_frameset.html" TARGET="_parent" onClick=" kilepes();"
" ...><IMG NAME="Kilépés" SRC="Navbar/Menugombok/navbar_b12.gif" BORDER="0"
ALT="Kilépés" ></A>
```

Az **analóg órát** a ThomasWebMűhely Javascript archívumából töltöttem le, majd a Navbar frame-be beillesztettem. A beillesztését követően pozicionáltam a helyét. Mivel az óra egymásra helyezett layerekből épül fel, ehhez a forráskódot négy helyen is korrigálnom kellett. Az egyes layereken az óra egy-egy „alkatrésze” található (külön layeren a számlap, a kis-, nagy- és másodpercmutató). Egységes áthelyezésükhöz minden layer top és left paraméterét ugyanarra az értékre (top:-455px, left:-85px) kellett állítani.

Nehézségként jelentkezett továbbá, hogy mivel a számlap számai külön-külön rétegeken vannak definiálva, - azaz ezek együttese adja a számlap egészét - előfordult, hogy az egyes layerek átfedték egymást, így a számok egyes részei nem látszódtak. A problémát a layerek „width” és „height” paraméterében meghatározott értékek csökkentésével orvosoltam, az alábbi forráskód alapján:

```
<div id="ieDigits" style="position:absolute; top:0px; left:0px; width:20px;
height:20px; font-weight:bold; font-family:Ariel; font-size:12px; color:
'+fCol+'; text-align:center; padding-top:5px">'+i+'</div>
```

A fenti kódrészlet „font-weight”, „font-family” és „font-size” paramétereit is beállítottam: a betűk (ti. számok) nagyságát 10-esről 12-esre növeltem, vastagságát „bold” értékre állítottam, típusát „Arielre”. Az így végrehajtott változásokkal az óra számlapja láthatóbbá, olvashatóbbá vált.

A módosítások hatására az analóg órát a Microsoft Internet Explorer böngésző tökéletesen kezelte, ám a későbbi tesztelés során kiderült, hogy más browserek (Opera 6.01, Netscape 6.1, Mozilla 1.0) nem értelmezik azt (azaz nem látható általuk az óra). A probléma gyökerét a következő feltételes elágazásban fedeztem fel:

```
ns = (document.layers)?1:0;
if (ns) { ... }
else { ... }
```

Azaz a program definiál egy „ns” nevű változót, amelynek a document.layer elemtulajdonság létezése esetén egy az értéke, nem létezésekor nulla. Ha ez az érték egy, az if ág fut le, ha nulla, akkor az else ág.

A feltételes elágazásra azért van szükség, mert a Netscape típusú böngészők és az Explorer típusú böngészők másként értelmezik a layereket. A tesztelésből azonban kiderült, hogy a program lefutásakor mindkét típusú böngésző az „ns” változót nulla értékkel tölti fel, ami az else ág lefutását eredményezi. Ez az ág azonban a Netscape 6.1 számára ismeretlen réteg definíciókat tartalmaz, s emiatt nem tudja az órát megjeleníteni.

A következő feltétel beépítésével próbáltam megoldani a problémát:

```
if (navigator.appName=="Netscape") { ns=1; }
```

Ebben az esetben bár lefutott az if ág, de a layereket sem a Mozilla 1.0, sem a Netscape 6.1 nem tudta helyesen értelmezni, mivel az óra számlapjának számai függőlegesen egymás alatt tűntek fel a képernyőn, a mutatókat pedig egyáltalán nem jelenítette meg.

A program működése megerősített abban, hogy a CD-ROM-ot Microsoft Internet Explorer böngészőre optimalizáljam, és emiatt töröljem az „ns” változót, valamint az összes feltételes elágazást, ami a változó meglététől függően bontotta két részre a programsort. Így aztán az Explorer típusú böngészőkben futni fog az óra, a Netscape típusúaknál pedig nem lassítja a letöltést, hiszen nem tartalmaz számukra értelmezhetetlen részeket.

E.) A sablonok

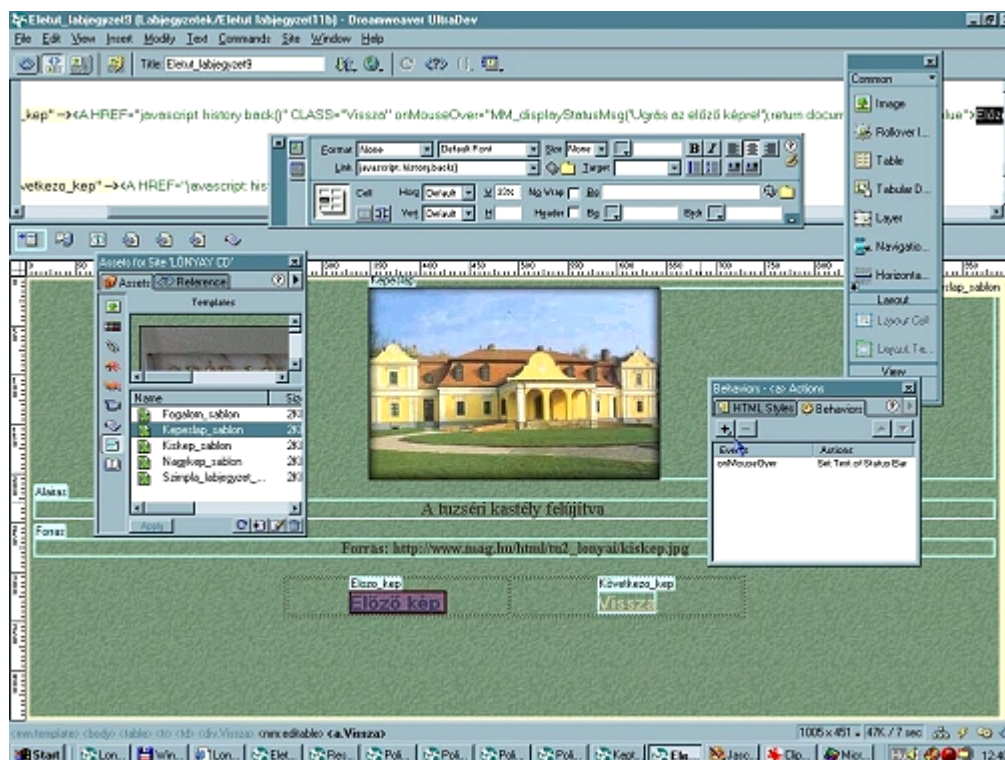
A HTML oldalak készítésének egyik kritikus pontja a hasonló szerkezetű oldalak előállítása. Ez a probléma nálam legélesebben a közel kilencszáz lábjegyzet-oldal elkészítésekor jelentkezett. A nehézség megoldását a Dreamweaver Templates (Sablonok) opciójának használata jelentette. Segítségével a már kész HTML oldal sablonként elmenthető (File→Save as Templates), amelynek bizonyos - általam meghatározott részei - szerkeszthetővé tehetők (Jobb klikk→New Editable Region), a többi elemei pedig öröklődve fixek lesznek. A sablonokat alkalmazhatjuk már meglévő HTML oldalak újraformázására, illetve új oldalak készítésére is.

Az utóbbi lehetőséget kiválasztva a lábjegyzetek készítése lényegesen felgyorsult. A lábjegyzetek készítése során a már sablonként elmentett és külön Templates mappában tárolt oldalak közül csupán kiválasztottam a szükséges sablonformátumot, annak szerkeszthető részeit a kellő adatokkal feltöltöttem (lsd. 3. számú képet), végül a megfelelő helyre elmentettem.

A lábjegyzetek sablonjait négy típusra osztottam. Az első típust ún. „szimpla lábjegyzet”-nek neveztem el, ez a csupán szöveget tartalmazó lábjegyzet-formátum (lsd. 4. számú mellékletet). A második, a „kiskep_sablon” nevű a szöveget és kisméretű képet tartalmazó lábjegyzet-sablon (lsd. 5. számú mellékletet), a harmadik (nagykep_sablon) az egy darab nagyobb méretű képet tartalmazó lábjegyzetek (lsd. 6. számú mellékletet), míg a negyedik (kepeslap_sablon) a képsorozatot tartalmazó oldalak formátuma (lsd. 7. számú mellékletet).

A sablonok használatával jelentős időt, energiát takarítottam meg, a monoton munkafolyamatot kicsit színesebbé tettem.

Hasonló technikát alkalmaztam a Segítség menüpont alá tartozó HTML oldalak elkészítésekor is.



3. számú kép Dreamweaver munkaoldal: a sablonok használata.

A lábjegyzetek készítését a Dreamweaver még egy beállításának használata gyorsította meg: az Edit→Find and Replace funkciója a naplórészlet azonos szavainak megkeresését tette lehetővé, ezáltal az adott szóra vonatkozó lábjegyzet-utalást a napló bármely előfordulási helyén link formájában definiálni tudtam. Használata során viszont figyelni kellett a napló régies szóalakjaira, a névírásaira illetve a többféle írásváltozatokra.

V. Tesztelés

A CD-ROM-ot négyféle böngészőprogramban teszteltem: Microsoft Internet Explorer 4.0., Netscape 6.1, Mozilla 1.0 és Opera 6.01. A tesztelés során a CD-ROM egységes megjelenítésének elérésére törekedtem, de a programok különböző konfigurációja miatt ezt az elvet csak részlegesen sikerült megvalósítanom.

A tesztelés során a következő problémákkal találkoztam:

A böngészők közül egyedül a Microsoft Internet Explorer érti az tag ALT paraméterét, annak ellenére, hogy ez a paraméter a HTML 4.0-ás szabvány szerves részét képezi.

Máshogyan értelmezik az egyes browserek a nyitóoldalon található „Böngésző típusa” nevű link mögött futó Javascript programot. Két böngésző a Microsoft Internet Explorer és az Opera (ha a program személyes beállításai szerint Operának kell neveznie magát) hívja meg helyesen a nevét és a verziószámát. A többi böngésző konfigurációjában nem állították át ezt a paramétert, így a Netscape 6.1 és a Mozilla 1.0 is Netscape 5-nek nevezi magát.

Az alapbeállítások közül az Opera 6.01 tévedése a legérdekesebb, ugyanis fordítottan értelmezi az onMouseOn és az onMouseOut eseménykezelőket.

A Netscape 6.1-es böngészőben az Vissza Javascript alkalmazás működése esetleges. Ebben az az érdekes, hogy a fent említett konfigurálásból kiindulva azt gondolnánk a Mozilla sem kezeli jól a history.back() függvényt. A gyakorlat azonban az ellenkezőjéről tett tanúbizonyságot! A program futásának blokkolásakor a Netscape a „Back” gombot sem működteti!

Az Opera 6.01 a <DIR> tag-et egyáltalán nem érti, míg a <BLOCKQUOTE> címkével máshogyan formáz, mint a fent említett három browser bármelyike.

A főoldalon lévő navigációs bár menüinek (ld. 2. számú melléklet) státuszszorban való kiíratását - a frame-es szerkezet miatt - a Mozilla 1.0 csupán a window.parent.status esetén alkalmazza, míg más böngészőkben a window.status esetén is megtörténik a kiíratás.

Eltérően értelmezik a böngészők a CD-ROM egyes formázóutasításait is: a szövegesen megadott betűnagyságok (pl. medium, large...), betűvastagságok (pl. bolder) nem mindenütt hajtottak végre. Az oldalak formázását ezért úgy alakítottam ki, hogy ezen beállításokat igyekeztem mellőzni! Az Opera 6.01 nem érti, a többi böngésző kicsit másképpen értelmezi az Időszalag menüpont Jump Menüjének formázását (betűtípus, betűméret, háttérszín stb.).

A felmerülő problémákra a forráskód szerkesztésével igyekeztem megoldásokat találni. A négy böngészőben való egységes megjelenítés érdekében ezért néhol többször kellett ugyanazt az utasítást kiadnom. Így például, a táblázatok középre zárását egy a stíluslapon középre definiált paragraph tag táblázat elé helyezésével próbáltam megoldani, de mivel ezt a Netscape 6.1 nem értette, a <TABLE ALIGN=CENTER> utasítást is ki kellett adnom. (Utóbbi tag bár benne van a HTML 4.0-ás szabványban, de a Word Wide Web Consortium (W3C) által nem javasolt eljárások közé tartozik. Alkalmazására a weblap működése érdekében került sor!)

A forráskód szerkesztése főleg az Opera 6.01 böngészőben eredményezett látványos változásokat. (Például az összes <DIR> tag eltávolításával!)

A tesztelés eredményeként - mivel a CD-ROM beállításait a legjobban adja vissza - az offline-weblap megtekintését elsősorban Microsoft Internet Explorer 4.0-ás böngészővel 1024*768-as képernyőfelbontás mellett ajánlom.

VI. Összegzés

Szakdolgozatom egy történelmi forrás HTML alapú feldolgozására tett kísérletet. Célja volt, hogy a tartalomhoz igazodva, az ízlésség, az élvezhetőség, a praktikusság és a használhatóság határain belül a HTML elemek és Javascript alkalmazások minél változatosabb tárat vonultassa fel; s mindemellett az, hogy egy régi, értékes és az utókor számára jól használható naplót sokoldalú szempontok alapján digitalizáljon.

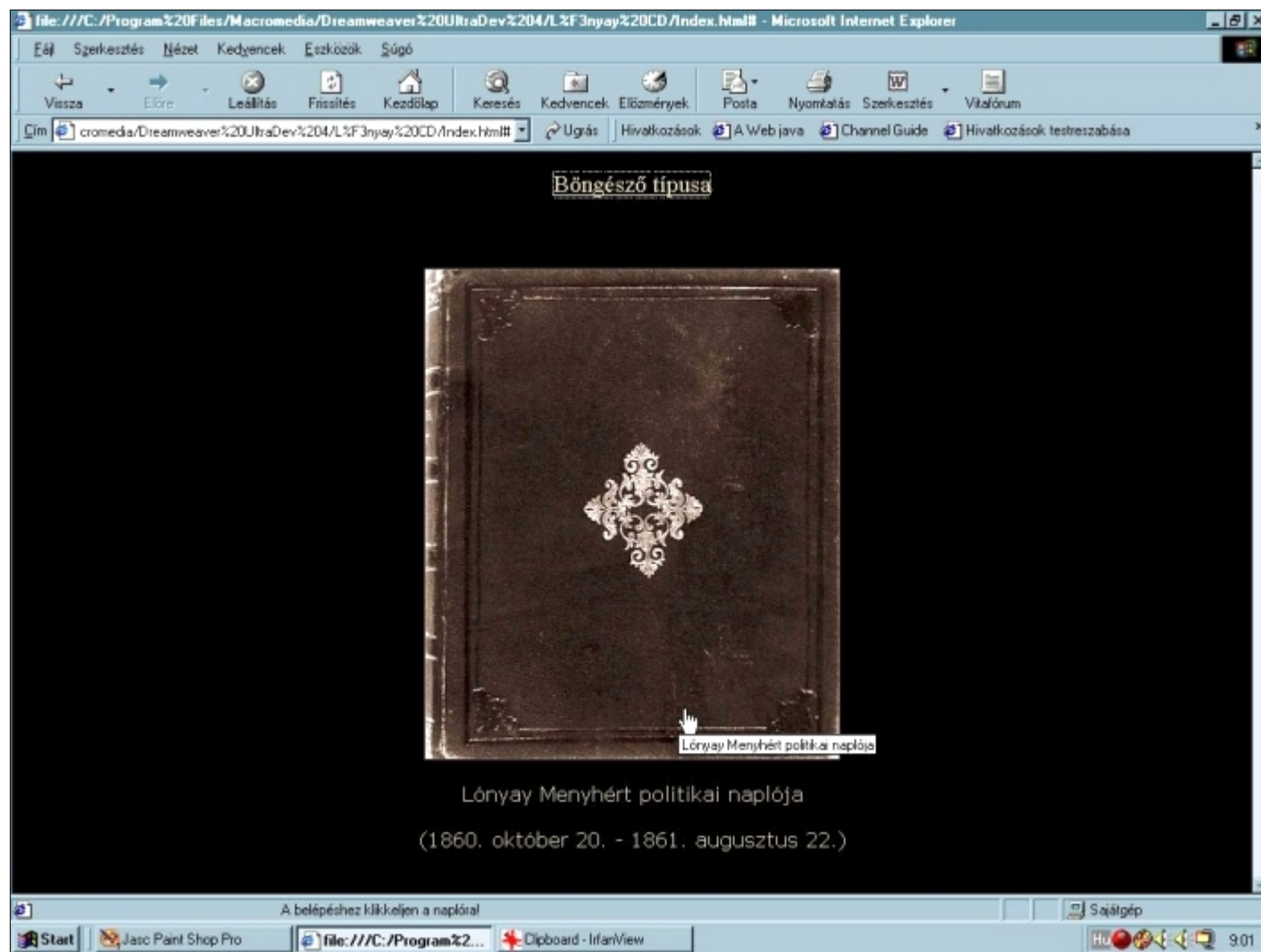
Célkitűzéseimet úgy hiszem a lehetőségekhez képest sikerült megvalósítanom, bár a képek gyűjtésével, minőségük javításával még élvezetesebbé lehetne tenni a forrás olvasását.

A CD-ROM-mal kapcsolatos további terveimben szeretném a forrást hiányzó részeivel kibővíteni, mivel ez a feldolgozás az 1860-61-es Lónyay-napló csupán egyharmadát dolgozza fel. (Ennek oka nem más mint, hogy a napló ezen részei álltak csak a rendelkezésemre.) A weblap kiegészítését a már fent lefektetett szempontok alapján tervezem, s ha a kiadásával kapcsolatban igény mutatkozik rá, szeretném is megvalósítani.

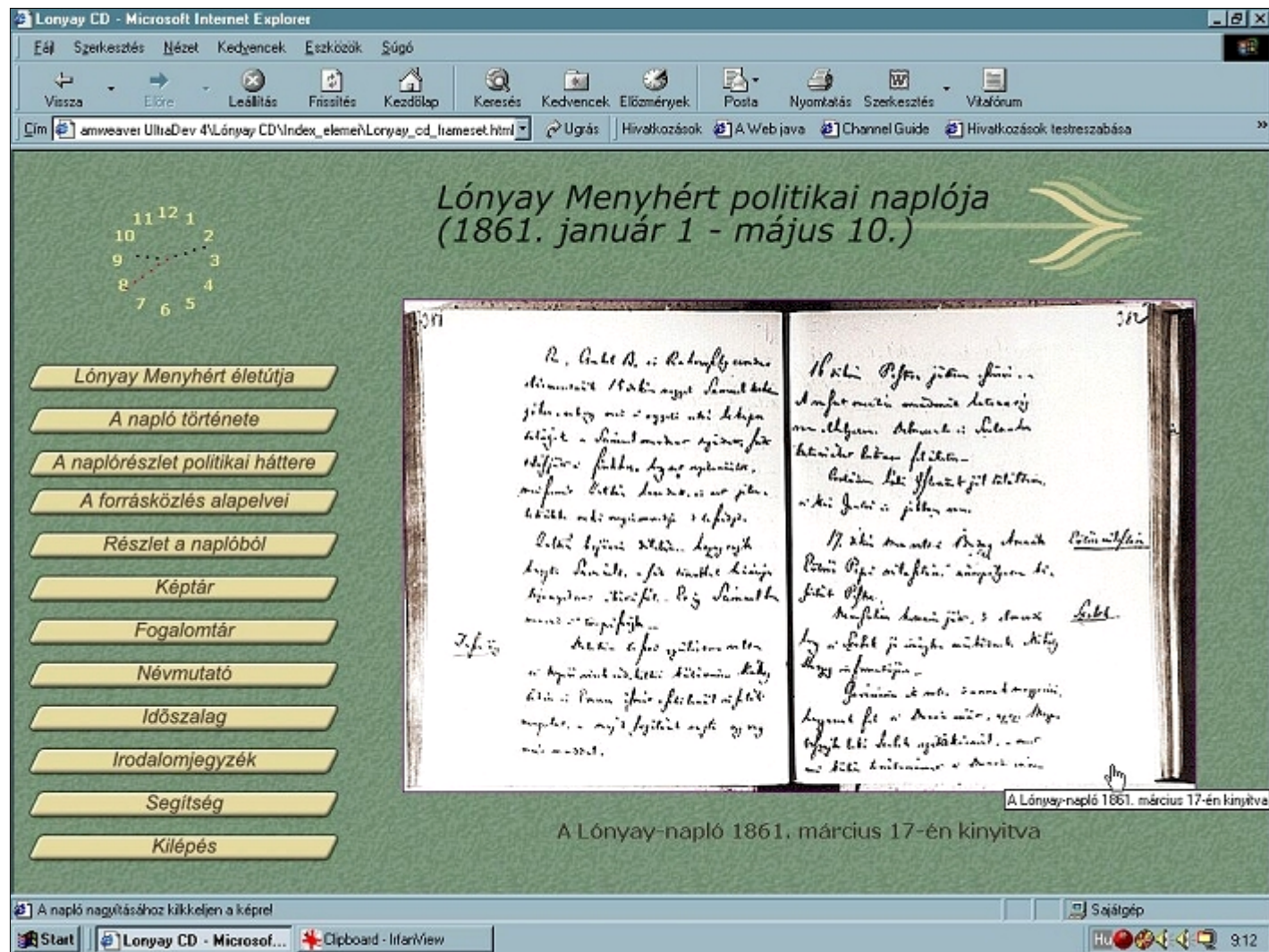
IRODALOMJEGYZÉK

- Bártfai Barnabás - Kovács Rudolf: Weblapkészítés házilag Dreamweaver című fejezet
Bp., BBS-INFO Kiadó, 2002.
- Bócz Péter - Szász Péter: A világháló lehetőségei. Interaktív weblapok készítése
Bp., ComputerBooks, 1998.
- Fodor Orsolya: Dreamweaver 4 kézikönyv
Bp., Horváth és Fellner Kft., 2001.
<http://web.axelero.hu/fodorsi/index.html>
- Galántai Zoltán - Komáromy Gábor: Web page saját kezűleg
Bp., Virgil Kiadó, 1997.
- <http://htmlinfo.polyhistor.hu>
- <http://www.intermedia.c3.hu/javascript/main.html>
- Kris Jamsa - Suleiman „Sam” Lalani - Steve Weakley: A Web programozása I-II.
Bp., Kossuth Kiadó, 1997.
- Kiss István: A Java testvérkéje a JavaScript
<http://www.cab.u-szeged.hu/WWW/java/Kiss>
- Péter Gábor: HTML 4.0 A világnyelv
Bp., University Press, 1998.
- ThomasWebMuhely Javascript archívuma
<http://mitglied.lycos.de/thomaswebmuhely/twm/js>
- Warner, Janine: Dreamweaver 4
Bp., Kossuth Kiadó, 2002.
- Warner, Janine - Vachier, Paul: Dreamweaver 4 Dummies könyvek
Bp., Kossuth Kiadó, 2002.

MELLÉKLETEK



1. számú melléklet: Az index.htm oldal



2. számú melléklet: A főoldal



3. számú melléklet: A nyomógombok



4. számú melléklet: Lábjegyzet 1.


Lónyay CD - Microsoft Internet Explorer

Élő Szerkesztés Nézet Kedvencek Eszközök Súgó


Vissza Előre Leállítás Frissítés Kezdőlap Keresés Kedvencek Előzmények Posta Nyomtatás Szerkesztés Vitafórum

Cím amweaver UltraDev 4\Lónyay CD\Index_elemei\Lónyay_cd_tartalom.html Ugrás Hivatkozások A Web java Channel Guide Hivatkozások testreszabása


Lónyay Menyhért politikai naplója (1861. január 1 - május 10.)



- Lónyay Menyhért életútja
- A napló története
- A napló részlet politikai háttere
- A forráskiadás alapelvei
- Részlet a naplóból
- Képtár
- Fogalomtár
- Névmutató
- Időszalag
- Irodalomjegyzék
- Segítség
- Kilépés



Eötvös József, br.(1813-1871): író, költő, centralista politikus, 1848-ban vallás- és közoktatásügyi miniszter, 1861-ben képviselő, a kiegyezés híve, Lónyay személyes, jó barátja; és valószínűleg



Eötvös Tamás (1800-1867): volt forradalmi kormánybiztos, országgyűlési képviselő. A reformkorban a beregi ellenzék vezére, országgyűlési követ. 1848-49 fordulóján Munkács város parancsnoka, majd Ung és Bereg megyék forradalmi kormánybiztosa. 1852-ben hatévi várfogságra ítélték. Az 1861-es és az 1867-es országgyűlések tagja.

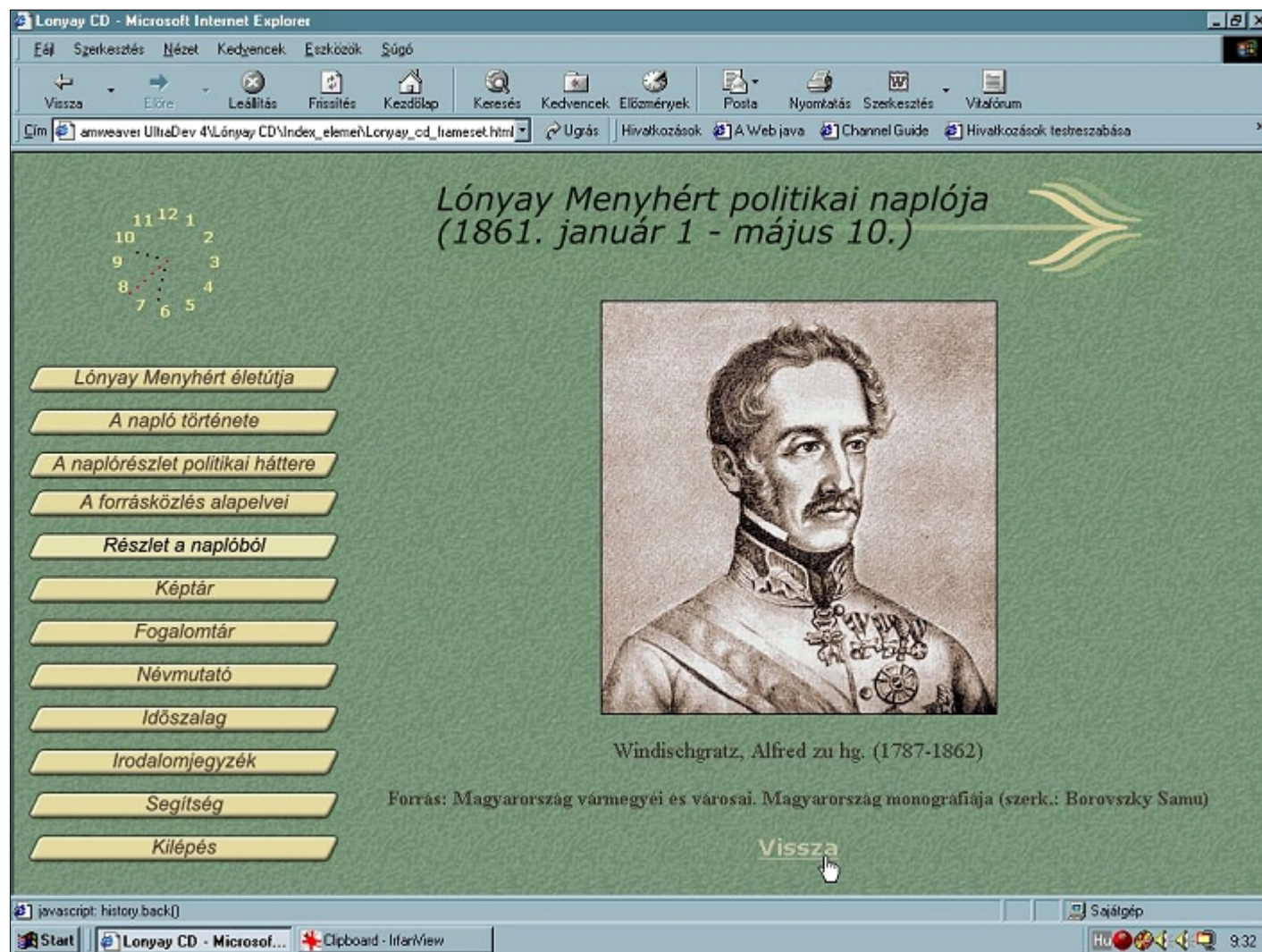
[Vissza](#)

Segítség Saját gép

Start Lónyay CD - Microsof... Clipboard - IfarView

9:20

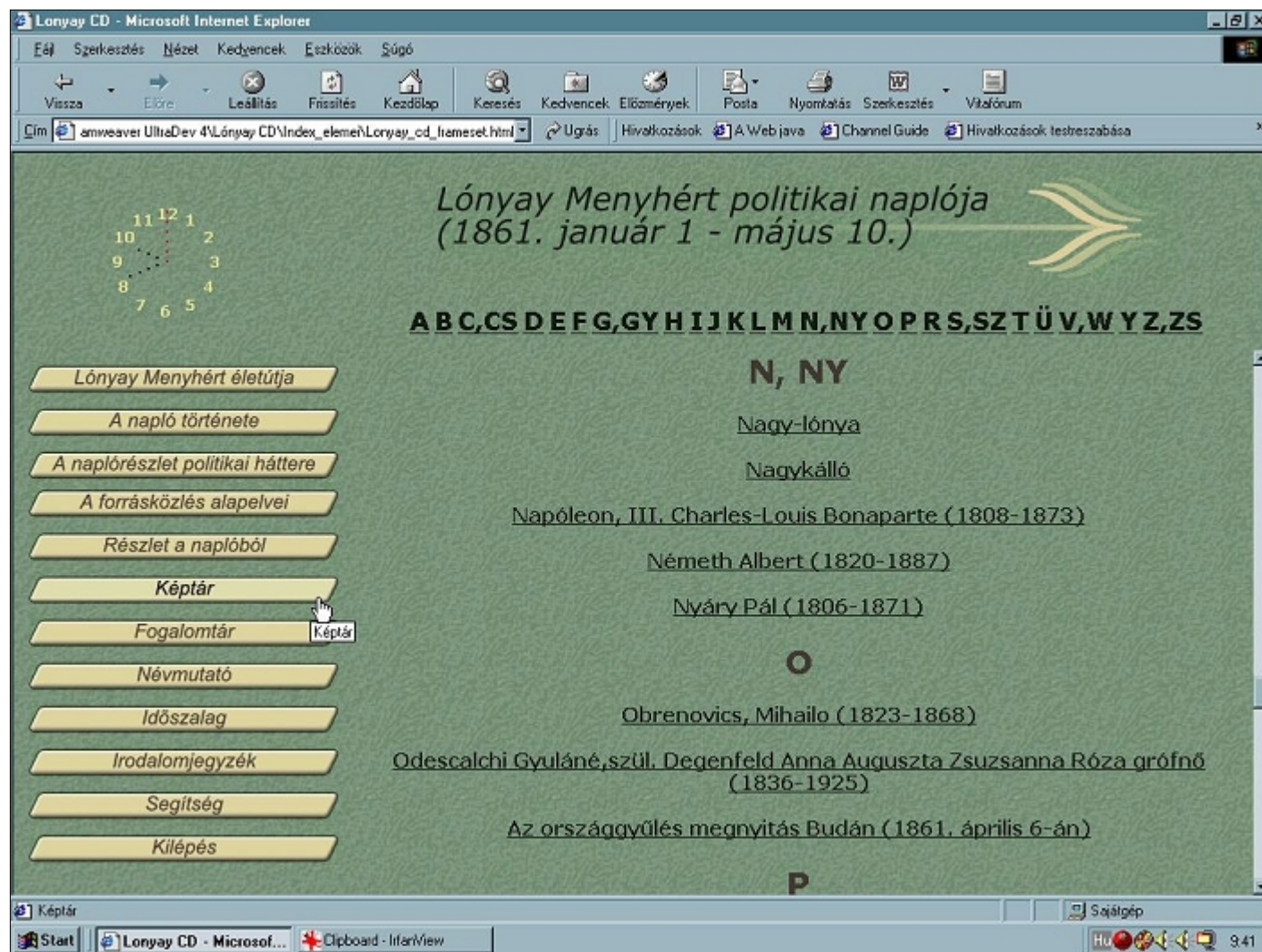
5. számú melléklet: Lábjegyzet 2.



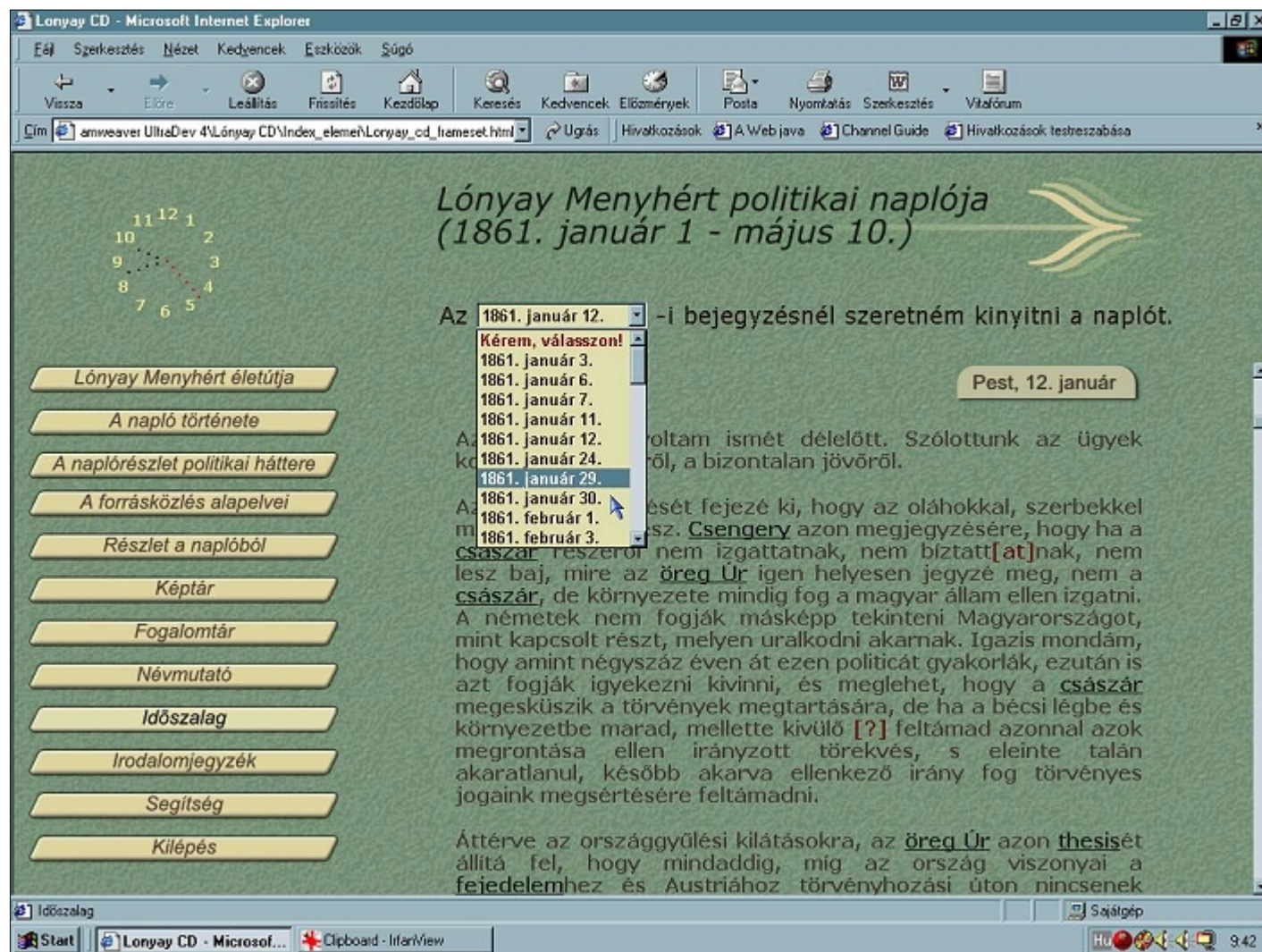
6. számú melléklet: Lábjegyzet 3.



7. számú melléklet: Lábjegyzet 4.



8. számú melléklet: A Képtár menü



9. számú melléklet: Az Időszalag menü



10. számú melléklet: Az Irodalomjegyzék menü



11. számú melléklet: A Kilépés menü