# Tartalomjegyzék

1. Bevezetés	3
2. Elvárások a Weblappal szemben	4
2. 1. Feladat specifikáció	4
2. 2. Tananyag	5
2. 3. Használati esetek	5
2. 3. 1. Felhasználók	5
2. 3. 2. Használati eset diagramm	6
2. 3. 3. Felhasználói esetek	7
2. 3. 4. Felhasználói esetek részletezése	8
2. 4. Beviteli űrlapok	9
3. Tervezés	.12
3. 1. Adatbázis terv	.12
3. 1. 1. Egyed – kapcsolat diagram	.12
3. 1. 2. Relációs adatbázis séma	.12
3. 1. 3. Az adatbázis táblái	.13
3. 2. Navigációs terv	.16
3. 3. Osztályok és függvényeik	.16
4. Kódolás	18
4. 1. Fejlesztő környezet	.18
4. 2. Szabványok figyelembe vétele	18
4. 3. A beléptető script	19
4. 4. PHP megoldások	19
5. Oktatási hatékonyság	21
5. 1. Oktatási stratégia	21
5. 2. Hasznosság és használhatóság	22
5. 3. Elméleti ismeretek	23
5. 4. Szemléltetés és ellenőrzés	24
6. Felhasználói dokumentáció	26
6. 1. Az Appserv telepítése	26
6. 2. Futtatás helyi és webes szerveren	30
6. 3. A phpMyAdmin használata	34

7. Tesztelés	38
7. 1. Üzembe helyezési teszt	38
7. 2. Funkcionális teszt	38
8. Összegzés	40
9. Irodalomjegyzék	41

# 1. Bevezetés

Napjainkban, az oktatásban a számítástechnikai eszközök egyre nagyobb teret foglalnak. Számos iskola alkalmazza az Internetet, hogy diákjaik otthonról is információkhoz juthassanak, valamint oktatási segítséget kaphassanak. Sok esetben egyszerűen csak az elérhetőségeket, valamint 1-2 oktatási segédletet tartalmaznak ezek az iskolai Weblapok. A kommunikáció tanár és diák között ritkább esetben jelenik meg, ez a funkció még hiányként mutatkozik a középiskolák honlapjain.

Ennek a Weblapnak összetett lesz a feladata, de csak a középiskolai oktatás egy részére specializálódik. Az oldal témája a középiskolák fizika tananyagának egy része, konkrétan az ütközések. A regisztrációnak köszönhetően a személyiségi jogok figyelembe lesznek véve, ez azt jelenti, hogy a hozzáférés a nem illetékes személyek részére nem lehetséges. Az információtovábbítás az üzenetek küldésével lesz megoldva, valamint a diákok megtekinthetik az értékelésüket az Interneten keresztül, ez a módszer a főiskolákon már elterjedt. A fizikaoktatásban jelentős probléma, hogy kevés esetben van rá lehetőség, hogy szemléltessék is az aktuális fizikai folyamatot. A számítástechnika eszközeivel ez is könnyen megoldható, ez esetben az ütközéses kísérletek bemutatására animációk fognak szolgálni. Másik probléma, hogy az otthoni megoldásra kiadott feladatokat nem tudják leellenőrizni a diákok. A Weblapos módszer erre is megoldás lehet, mivel a feltöltött példák megoldása után a tanuló megnézheti, hogy az általa kiszámolt eredmények helyesek-e.

Mivel Weblapról van szó, a választott megoldás a HTML és PHP programozási nyelvek együttes alkalmazása valamint MYSQL adatbázis használata. A választás oka, hogy e két nyelvet egyre több szolgáltatónál lehet alkalmazni, valamint az adatbázisok körében igen elterjedt a MYSQL alapú adatbázis kezelés, nem is szólva arról, hogy a MYSQL ingyenesen beszerezhető. A PHP nyelv mellett szólt az is, hogy nagyon sokféle megoldási lehetőség adódik Web programozás közben. Sokrétű adatbázis kezelést tesz lehetővé, valamint a hozzáférés korlátozása is megoldható.

A tervezésnél fontos szempontként figyelembe kell venni, hogy a Weblap felhasználó közeli és könnyen kezelhető legyen. A cél az, hogy megfeleljen egy középiskola igényeinek, valamint az oktatási témakörrel szemben elvárt kritériumoknak. Több osztály és példa típus kezelhető legyen, lehessen regisztráció nélkül is elméleti ismereteket szerezni, valamint a kommunikáció se jelentsen problémát.

### 2. Elvárások a Weblappal szemben

#### 2. 1. Feladat specifikáció

A Weblap alapvető feladata, hogy műszaki középiskolákban a Fizika tantárgy *Ütközések* témakörének oktatását segítse. Ennek megfelelően tartalmazza az elméleti ismereteket, valamint ütközési szimulációkat mutat be. Ezek mélysége és tartalma a NATtól függ. A Weblap, ezen kívül tartalmaz egy adatbázist, amely a kővetkezőket valósítja meg.

#### Az Admin lehetőségei

Lehetősége van arra, hogy az adatbázis összes táblájához hozzáférjen, legyen az akár regisztráció is, és ha igény van rá, akkor módosítás is lehetséges. A módosítások és a törlések a későbbi fejlesztés során kerülnek át Webes felületre, addig az Adminnak hozzáférése van az adatbázishoz, ahol az esetleges változásokat meg tudja tenni, valamint a hibás bejegyzéseket tudja törölni.

#### A Diákok lehetőségei

A diákok belépés nélkül megtekinthetik a Tanárok által feltöltött példákat, ezek megoldását is ellenőrizhetik, valamint az elméleti ismereteket és szimulációkat is megnézhetik. Belépés után lehetőségük van arra, hogy megtekintsék az érdemjegyeiket, valamint a Tanár által küldött üzeneteket.

### A Tanárok lehetőségei

A tanárok a belépés után feltölthetnek példát a tananyaghoz kapcsolódóan, amely áll egy szöveges részből, ami a feladat maga, valamint a várható eredményeket (maximum 5-öt), amely segítségével a diákok ellenőrizhetik az általuk megoldott példát. Jegyeket is tölthetnek fel az általuk tanított osztályok diákjainak. Ezeket és a korábbi jegyeket maguk a tanárok is meg tudják nézni. Lehetősége van üzeneteket küldeni az egész osztálynak, a későbbiekben adott diáknak is.

#### 2. 2. Tananyag

A középiskolai fizika oktatás széles témakörének egy részét hivatott ez a Weblap tartalmazni. A Dinamika témakörön belül, a Lendület megmaradás törvényéhez tartozik a fizikai ütközések problémája. A célkitűzés az, hogy egy olyan Internetes oldal készüljön, ami otthon is segítséget nyújthat a tanulásban. Ha elakad a diák, akkor segítséget kaphat az elméleti anyagból, ellenőrizheti tudását a feltöltött példák segítségével, valamint megtekintheti osztályzatait és a tanárok által küldött üzeneteket.

Az ütközések témakör elméleti ismereteit középiskolai fizika könyvek alapján érdemes bemutatni. Ezek a könyvek megfelelően tartalmazzák az ütközések témaköréhez tartozó fontos ismereteket [1, 34-35. oldal]. Számos oktató oldal is tartalmazza azon ismereteket, amelyekre szükség lehet ahhoz, hogy a témakörhöz szükséges tudást megszerezhessék a tanulók. Ezen oldalak közül a www.sulinet.hu oldal egyik cikke pont megfelelő arra, hogy a kellő ismeretekhez hozzáférjen az ember [2].

### 2. 3. Használati esetek

[3, 14-23. oldal]

### 2. 3. 1. Felhasználók

Ezen Weblap esetében 3 felhasználó típus fordul elő, mindegyiknek más és más lehetőségei és hozzáférési jogai vannak. A legfőbb Aktor az *Admin* akinek, mindenhez hozzáférése van, valamint a többi felhasználó felvételét is ő végzi. A másik két user egymással kapcsolatban van, de viszonyuk alá-fölérendelt. A *Tanár* változtathat a *Diák*okhoz tartozó adatokon, míg a *Diák* csak megtekintheti ezeket. Fontos, hogy a személyes információkhoz csak arra jogosult személy juthasson, ezért mindegyik felhasználó típus a jelszavas beléptető rendszeren keresztül tud belépni.

# 2. 3. 2. Használati eset diagramm



2. 3. 2. a. Ábra. Használati esetek diagram

# 2. 3. 3. Felhasználói esetek

Tanár/Diák regisztráció:	Az Admin itt adhat hozzá egy új		
	felhasználót, aki lehet Tanár vagy Diák.		
Elmélet feltöltése:	Az Admin ez esetben megszerkesztheti		
	az elméleti ismereteket.		
Elmélet módosítása:	Az Admin ez esetben meg tudja		
	változtatni az elméleti ismereteket.		
Módosítás/Törlés:	Az Admin itt az adatbázishoz fér hozzá,		
	és végrehajthat módosítást és törlést.		
Példa feltöltése:	A Tanár új példát adhat hozzá az		
	adatbázishoz.		
Jegy feltöltése:	A Tanár itt új jegyet tölthet fel az általa		
	tanított osztályok diákjainak.		
Jegyek lekérdezése (Tanár):	A Tanár itt megtekintheti az általa tanít		
	diákok jegyeit.		
Üzenet küldése:	A Tanár üzenetet küldhet az osztályainak.		
Elmélet megtekintése:	A Diák megtekintheti az elméleti		
	ismereteket valamint a kísérleteket.		
Példa megtekintése/ellenőrzése:	A Diák megtekintheti a példákat típus		
	szerint, valamint leellenőrizheti, hogy		
	helyesek-e a megoldásai.		
Jegyek lekérdezése (Diák):	A Diák lekérdezheti saját osztályzatait.		
Üzenetek olvasása:	A Diák elolvashatja az osztály üzeneteit.		

### 2. 3. 4. Felhasználói esetek részletezése

Itt a magyarázatra szoruló felhasználó eseteket kerülnek bemutatásra, amelyek címe félreértésre adhat okot.

Tanár/Diák regisztráció:

- Az Admin kitölti a regisztrációs űrlapot
- Ellenőrzés: minden adat megvan-e, van-e már ilyen felhasználó
- Elküldés után az adatok feltöltése a Tanár illetve Diák táblába

Példa feltöltése:

- A Tanár kitölti az űrlapot: típus, szöveg, eredmények
- Ellenőrzés: minden adat megvan-e, legalább egy megoldást meg kell adni
- Elküldés után az adatok feltöltése a Példa és Eredm táblába

Példa megtekintése/ellenőrzése:

- Az adott típusú példák közül a Diák kiválasztja a példa sorszámát
- A példa elolvasása és megoldása után az eredmények kitöltése
- Elküldés után az Eredm tábla alapján a megoldások kiértékelése

Jegy feltöltése:

- A Tanár kiválasztja az osztályt, ahova jegyeket kíván feltölteni
- A Diák kiválasztása és az osztályzat megadása (csak megfelelő érték lehet)
- Elküldés után az adatok feltöltése a Jegyek táblába

Jegyek lekérdezése (Tanár):

- A Tanár kiválasztja a lekérdezni kívánt Diák osztályát, majd a Diákot
- Lekérdezés az adott Diák azonosítója alapján

Jegyek lekérdezése (Diák):

- Az adott Diák azonosítója alapján a Jegyek tábla megfelelő sorainak listázása

Üzenet küldése:

- A Tanár kiválasztja az általa tanított osztályok közül a címzettet
- Megírja a Tanár az üzenetet
- Elküldés után az üzenet tárolása az Uzenet táblában

Üzenet olvasása:

- Az adott Diák osztályának küldött üzenetek megtekintése

# 2. 4. Beviteli űrlapok

A felhasználói felületek tervezésekor fontos szempont, hogy a kitűzött célok eléréséhez megfelelően alakítsuk ki. Az egyik legfontosabb tervezési rész a beviteli űrlapok, mivel ezeken keresztül férhetnek hozzá a felhasználók az adatbázishoz. A tervezésnél figyelni kell, hogy minden olyan adatot bekérjen az űrlap, amire szükség van az adatbázisban. Az űrlapokat 3 csoportra lehet osztani, amely csoportok megegyeznek a felhasználói típusokkal.

# Admin űrlapjai

Osztály:	Osztály 💌
Név:	
Születési dátum:	
Lakcím:	
Jelszó:	
Jelszó még egyszer:	

# Diákok regisztrációja

Regisztráció

# Tanárok regisztrációja

Osztály:	Osztály 💌
Név:	
Jelszó:	
Jelszó még egyszer:	

Regisztráció

# Tanárok űrlapjai

### Példa feltöltése

Típus 💌

Példa típusa:

Példa szövege:



Megoldások a feladatra:

1.	<u>.</u> ]	2.	٨.	5.	
1:	2:	 3:	4:	5.	

Feltöltés

Üzenetküldés

Osztály:

Osztály 💌

Üzenet szövege:



Küldés

# Jegy feltöltése

Diák neve: Név 🔽 Osztályzat:

Feltöltés

# Diákok űrlapjai

# Példa megoldása

Itt olvasható a példa szövege NEM beviteli mező.

Megoldások a feladatra:



Tovább

# 3. Tervezés

# 3. 1. Adatbázis terv

# 3. 1. 1. Egyed- kapcsolat diagram

[4]



# 3. 1. 1. a. Ábra

# 3. 1. 2. Relációs adatbázis séma

TANAR (<u>TID</u>, osztály, nev, jelszo )
DIAK (D<u>ID</u>, osztály, nev, szdatum, lakcim, jelszo )
JEGYEK (*DID*, *datum*, *ertek*)
ADMIN ( nev, jelszo)
UZENET (osztaly, datum, szoveg)
PELDA (<u>PID</u>, tipus, szoveg)
EREDM (*PID*, a, b, c, d, e)

# 3. 1. 3. Az adatbázis táblái

A FIZIKA adatbázis 7 táblát tartalmaz, melyek a következők.

# TANAR tábla:

Oszlopnév	Oszloptípus	Hossz	
TID	INT	2	,
osztaly	VARCHAR	4	-
nev	VARCHAR	30	)
jelszo	VARCHAR	8	;

Ez a tábla a Tanárok adatait tartalmazza, akik tanítják az adatbázisban szereplő osztályokat A TANAR tábla tartalma:

TID:	a Tanár azonosítója
osztaly, osztaly2:	a Tanár által tanított osztályok
nev:	a Tanár neve
jelszo:	a Tanár jelszava

# DIAK tábla:

Oszlopnév	Oszloptípus	Hossz
DID	INT	2
osztaly	VARCHAR	4
nev	VARCHAR	30
szdatum	date	
lakcim	VARCHAR	30
jelszo	VARCHAR	8

Ez a tábla a Diákok adatait tartalmazza, akik az Ütközések témakörét tanulják

A DIAK tábla tartalma:

DID:	a Diák azonosítója
osztaly:	a Diák osztálya
nev:	a Diák neve
szdatum:	a Diák születési dátuma
lakcim:	a Diák lakcíme
jelszo:	a Tanár jelszava

### ADMIN tábla:

Oszlopnév	Oszloptípus	Hossz
nev	VARCHAR	30
jelszo	VARCHAR	8

Ez a tábla a Rendszergazda adatait tartalmazza.

Az ADMIN tábla tartalma:

nev:	az Admin neve
jelszo:	az Admin jelszava

### PELDA tábla:

Oszlopnév	Oszloptípus	Hossz
PID	INT	2
tipus	VARCHAR	15
szoveg	VARCHAR	255

A PELDA tábla a példák tulajdonságait tárolják.

A PELDA tábla tartalma:

PID:	a Példa sorszáma
tipus:	a Példa típusa
szoveg:	a Példa szövege

# JEGYEK tábla:

Oszlopnév	Oszloptípus	Hossz
DID	INT	2
datum	date	
ertek	int	1

Ez a tábla a diákok azonosítója alapján tárolja a jegyeket.

A JEGYEK tábla tartalma:

DID:	az adott jegyhez tartozó Diák azonosítója
datum:	a jegy feltöltés dátuma
ertek:	a jegy értéke

# EREDM tábla:

Oszlopnév	Oszloptípus	Hossz
PID	INT	2
а	VARCHAR	15
b	VARCHAR	15
С	VARCHAR	15
d	VARCHAR	15
е	VARCHAR	15

Ebben a táblában a példákhoz tartozó eredményeket tárolják.

Az EREDM tábla tartalma:

PID:	a Példa sorszáma, amelyhez az eredmény tartozik
a :	a Példa első megoldása
b:	a Példa második megoldása
c:	a Példa harmadik megoldása
d:	a Példa negyedik megoldása
e:	a Példa ötödik megoldása

# UZENET tábla:

Oszlopnév	Oszloptípus	Hossz
osztaly	VARCHAR	4
datum	date	
szoveg	VARCHAR	255

Az UZENET tábla tartalma:

osztaly:	a címzett osztály azonosítója
datum:	az üzenet küldés dátuma
szoveg:	az üzenet szövege

# 3. 2. Navigációs terv





# 3. 2. a. Ábra

# 3. 3. Osztályok, függvényeik és feladataik

- A Login osztály: feladata a beléptetés és a nyomon követés
  - konstruktor: kezdőértékeket ad és figyeli, hogy jött e kilépési kérelem

- beleptet (): ellenőrzi, hogy a kapott azonosító helyes-e, akár az űrlapból jött, akár a session-ből. Ha nem kapott azonosítót, akkor a beléptető űrlapra irányít. Ha kapott azonosítót és az helyes, akkor eltárolja az azonosítót a session változóba.

- kileptet (): kilépési kérelem esetén törli a session változót, és a főoldalra irányít.
- loginUrlap() : amennyiben még nem történt belépés, akkor ez a függvény fogadja a belépéshez szükséges adatokat (név, jelszó)

A diakok osztály: feladata a diákokkal való adatbázis műveletek

- konstruktor: létrehozza az adatbázissal a kapcsolatot és ellenőrzi, hogy a regisztrációkor megadott név szerepel-e már az adatbázisban

- reg(): feltölti az új felhasználó adatait, majd megjeleníti őket

A tanarok osztály: feladata hasonló a diakok osztállyal, de van még egy függvény

- osztleker(): lekérdezi, hogy az adott Tanár melyik osztályokat tanítja

A peldak osztály: a feladat a pelda és eredm táblákkal való műveletek

- konstruktor: létrehozza az adatbázissal a kapcsolatot
- tolt (): feltölti az űrlapból kapott adatokat a táblákba, majd megjeleníti őket
- tipleker(): a kiválasztott típus alapján lekérdezi a megfelelő példákat
- pkiir(): megjeleníti a típusnak és sorszámnak megfelelő példát

 pell(): az űrlapból kapott adatokat összehasonlítja az adatbázis adott sorával, majd kiértékeli a kapott adatokat

A jegyek osztály: feladat a jegyek táblával való adatbázis műveletek

- konstruktor: létrehozza az adatbázissal a kapcsolatot

- diaklek (): a megadott név alapján lekérdezi az adott diák osztályzatait és megjeleníti őket

 listat(): lekérdezi az adott osztály diákjait és az adott diákhoz történő jegyfeltöltéshez ad beviteli mezőt

- listad (): lekérdezi az adott osztály diákjait és továbbadja a lekérdezni kívánt diák nevét.

 - jegytolt (): az űrlapból kapott adatok alapján feltölti az adatokat jegyek táblába megfelelő DID alapján.

Az uzenetek osztály: feladata az uzenet táblával való adatbázis műveletek

- konstruktor: létrehozza az adatbázissal a kapcsolatot

- kuld (): a tanár által tanított osztályoknak írt üzenetek feltöltése az űrlapból kapott adatok alapján

- leker (): az adott diák osztályának küldött üzenetek lekérdezése
- kiir (): a kapott osztály szerinti üzenetek kiíratása

# 4. Kódolás

[6]

# 4. 1. Fejlesztő környezet

A kódoláshoz a 1stpage2000 ingyenesen letölthető Weblap szerkesztő programot használtam, amely segítségével a HTML elemek használata jelentősen támogatott volt, Valamint a PHP nyelv szintaktikája is jobban áttekinthető. A program letölthető az Evrsoft oldaláról: http://infinity2.dnsprotect.com/~evrsoft/1stpage2.zip

Fejlesztés közben az igen közkedvelt és ingyenes Appserv v2. 4. 0. programot használtam helyi szervernek, amely letölthető az alábbi címről:

http://prdownloads.sourceforge.net/appserv/appserv-win32-2.4.0.exe?download

Az Appserv egy programcsomag, amely tartalmazza a következő elemeket:

- Apache 1. 3. 29
- PHP 4. 3. 6
- MySQL 4. 0. 18
- Zend Optimizer 2. 5. 1
- phpMyAdmin 2. 5. 6
- Perl 5. 8. 0

Ezen program előnye, hogy minden összetevőt tartalmaz, amelyre a fejlesztés során szükségem volt, ingyenes és bárki számára hozzáférhető. Valamint telepítéskor nem kell a konfigurálással bajlódni, ezért ha a Weblapot helyi szerveren működtetik, akkor könnyen üzembe helyezhető.

### 4. 2. Szabványok figyelembe vétele

A Weblap az XHTML 1. 0. szabvány alapján kerül szerkesztésre. Napjainkban egyre jobban elterjed ez a szabvány, ezért fontos, hogy e szerint készüljön el a Weblap. Újdonság és egyben nehézséget okozhat az új referenciákat használni, itt gondolok például a sortörés <br> tag-re, amely XHTML szerint így helyes: <br />. A többi "újítással" nem nagyon lehet gond, validáláskor a legtöbb esetben ez szokott a hiba lenni. A helyesség ellenőrzéséhez a http://w3c.org oldal használható, amellyel nem csak a HTML kódot, hanem a CSS elemekkel készített design-t is lehet ellenőrizni.

A CSS használata új keletű dolog a Weblap készítésben, de ez az újdonság jelentősen megkönnyítheti a későbbi fejlesztést. Rövid időn belül kiderülhet, hogy a segítségével hamar és könnyen át lehet alakítani a Weblap design-át. A CSS esetében is ügyelni kell a helyességre, ennek az ellenőrzésére is a már fentebb említett oldalt lehet használni.

### 4. 3. A beléptető script

[7]

A beléptetésre és a nyomonkövetésre a www.prog.hu Internetes oldal *Felhasználók* beléptetése és követése cikket használtam

Ez a script egy olyan kód, amely objektum orientáltan egy Login nevű osztállyal kezeli a felhasználókat. A megoldás tetszetős, mivel az ellenőrzés egyszerű de mégis hatékony. Ellenőrzi, hogy honnan jöttek a beléptetéshez szükséges adatok (űrlap vagy session), ha megfelelő az adat, akkor tárolja és beléptet, amennyiben nem, akkor a beléptető űrlapra irányít. Néhány módosítást kellett végezni, hogy a script ezen Weblap igényeinek is megfeleljen. Egyrészt általánossá kellett módosítani a megfelelő működéshez, a beleptet () függvény kapott egy paramétert, amely arra hivatott, hogy mindhárom felhasználó típus esetében a megfelelő táblából végezze az ellenőrzést, ez a paraméter az SQL lekérdezésben jelenik meg. Egy másik rész is módosításra szorult, mégpedig a session\_start() függvény került át, a login.php file végéről az elejére. A hiba, ami miatt erre szükség volt, az úgy jelentkezett, hogy belépési kérelemkor beléptetett, de a következő továbblépéskor nem "tudta" a \$HTTP\_SESSION\_VARS [] értékét, mivel csak a file végén került sor a munkamenet indítására.

Ezzel a beléptető megoldással egyszerűen megoldható minden olyan rész, amelyet csak jogosultsággal lehet megtekinteni, mivel elég egy pár soros script minden beléptetéssel működő file kódjának az elejére és máris megoldott a hozzáférési jogok ellenőrzése.

### 4. 4. PHP megoldások

Itt két megoldás kerül megemlítésre, az egyik a születési dátum bevitele, a másik, pedig a példák ellenőrzése.

Weblapoknál fontos szempont a figyelmetlen felhasználók elleni védelem, konkrétabban az, hogy azon adatok esetében, amikor fontos a formátum, akkor ne hibásan adja meg az adatokat, mert akkor az adatbázisban nem fog ez az adat megjelenni. Ezt a hibát úgy a legegyszerűbb kiküszöbölni, ha <select> tagban választhat a felhasználó, mert

akkor csak olyan adatot küldhet el, amit mi engedélyezünk. Az egyik ilyen probléma a Diákok regisztrációjánál lépett fel, mégpedig a születési dátum megadásánál. Először egy egyszerű beviteli mezőbe kellett bevinni az adatot, de fejlesztés során kiderült, hogy ha nem megfelelő a dátum formátum, akkor a MYSQL adatbázisban nem jelenik meg a kívánt dátum.

A megoldás az, hogy a születési dátumot 3 <select> tag oldja meg. Mindhárom beviteli mező esetében egy for ciklus állítja be, hogy milyen lehetőségek közül választhat a felhasználó. A lehetőségek egyértelműek, az évnél 1985-től az éppen aktuális évszámig írja ki a választási lehetőségeket. A hónapnál 12 és napnál 31 választási lehetőség van. A három megadott értéknek olyan neve van, hogy továbbküldés után egyszerűen összefűzhető legyen a megfelelő dátum formátumba, amit a MYSQL adatbázis DATE típusa már fel tud dolgozni, és tárolni tud.

A második megoldás, amiről szót szeretnék ejteni az a feltöltött példák ellenőrzése. Kezdetben az eredmények DOUBLE típusként lettek eltárolva, de fejlesztés közben előtérbe került egy lehetőség, hogy szöveges válasz is lehetséges. Ezért az EREDM tábla típusai VARCHAR ra módosultak, 15 karakteres szóhosszra. Feltöltéskor nem kötelező mind az 5 eredményt megadni, csak annyit amennyire szükség van. A feltöltés sikerességének kiíratásakor csak azok az eredmények jelennek meg, amelyek ténylegesen fel lettek töltve, tehát értékük van. Például: if(\$\_POST[a]!=""). Amikor a diák számára megjelennek a példák sorszámai, akkor az ezt végző függvény egy paramétert kap, amellyel a példa típusát adja meg, listázáskor csak a megadott típusú feladatok sorszámai jelenek meg. Kiíratáskor a már fentebb említett ellenőrzés van alkalmazva, miszerint csak azokra az eredményekre ad ki beviteli mezőt, amelyeknek értéke van az EREDM táblában. Ezután az adatbázisból lekérdezett, és az űrlapból kapott adatok összehasonlítása történik meg, szintén csak ott, ahol van érték. A megoldás értékelésekor megjelennek az adatbázisból, és az űrlapból kapott értékek is, valamint az értékelés, hogy a megoldás helyes e vagy helytelen, esetleg üresen maradt a beviteli mező. Az első próbálkozáskor a DOUBLE típus miatt, csak az volt figyelve, hogy nagyobb legyen, mint 0, de ez megváltozott, amikor az eredmény típus karakterláncra változott.

# 5. Oktatási hatékonyság

### 5. 1. Oktatási stratégia

Fizikaoktatásnál fontos, hogy az adott témakört kísérletekkel is szemléltessék, mivel ez pontosabb képet alakít ki a tanulóban az adott témában. Egy ábrán, de inkább egy kísérleten keresztül a törvények és képletek magyarázata is jelentősen egyszerűbb és érthetőbb. A kísérlet bemutatásánál ajánlott úgy jelölni a hatásokat, ahogy utána a törvényekben illetve a képletekben szerepelnek. Ezt a módszert követve alakult ki a következő stratégia.

### Témakör címe

- Az ütközések témakör rövid általános ismertetése

Az ide tartozó fizika alaptörvények bemutatása (lendület, energia-megmaradás törvénye stb.)

- A kísérletek bemutatása szimulációkon keresztül. Innen az oktatás két irányban halad tovább, a két rész tartalmilag különböző, de a tanítási módszer ugyanaz.

### Rugalmas ütközések

#### Rugalmatlan ütközések

- Szimulációk megtekintése
- Az elmélet magyarázata az ütközések típusa szerint
- A különböző típusú kísérletek magyarázata, a szimulációk alapján
- A szimulációban is szereplő képletek bizonyítása, magyarázata
- Végül a megoldáshoz szükséges egyenletek bemutatása
- Következmény levonása, amely bizonyítja az elmélet helyességét

#### 5. 2. Hasznosság és használhatóság

A Weblap célja már röviden ismertetve lett a bevezetőben, most részletesebben bemutatásra kerül, hogy miért készült ez a Weblap és miért így. Mint ismeretes napjainkban az Interneten egyre nagyobb lefedettség van a különböző területeken, gondolhatunk itt a vásárlásra, rendelésre stb. Ez a fejlődés megjelenik az oktatás világában is, mivel egyre több iskolának van Weblapja, amelyek oktató jellegűek. Az elméleti általános ismeretek már egy jó ideje, mint minden más téma, megjelentek az Interneten, kezdetben csak egyszerű HTML-es formában. Ahogy fejlődtek a technikák megjelentek az adatbázis háttérrel rendelkező oldalak. Ez már tágabb lehetőséget biztosít oktatási szempontból. Gondolhatunk itt a regisztrációra, ami a személyiségi jogokat védi, nem férhet hozzá bárki a személyes információkhoz, például üzeneteknél, amit csak egy bizonyos tag részére küldtek. Ezen Weblap is erre a módszerre épít, miszerint oktatói oldal, amely egy adatbázist is tartalmaz.

Fizika tanulmányai során mindenki találkozhat azzal a problémával, amit ez az oldal hivatott kiküszöbölni. Történetesen, hogy sok esetben nem tudják megfelelően bemutatni a kísérleteket, fizikai mozgásokat. A rajzolt ábrák, amelyek nyilakkal próbálják bemutatni a fizikai mozgásokat, nem minden esetben egyértelműek. Ezért folyamodtam a kísérletek szimuláción keresztül történő bemutatásához. Ezek az animációk Flash-ban készültek.

Egy másik érv, ami a Weblap haszna mellett szól, hogy egy MYSQL adatbázis van mögötte. Manapság viszonylag sok otthonban van Internet hozzáférés, amely tényt érdemes kihasználni oktatási célokra. Sok diáknak nehézséget okoz a fizika házi feladat megoldása, mert egyrészt hiányos a rendelkezésére álló tananyag, vagy pedig nem tudja elképzelni a fizikai ütközést. Ezekre a problémákra próbál megoldást adni a Tananyag menüpont, mely tartalmazza a témához kapcsolódó törvényeket és képleteket. Másik probléma szokott lenni, hogy nem tudja leellenőrizni a tanuló a megoldás helyességét, ezért vannak a Weblapon példák is, amelyek az önellenőrzésre használhatók, ezeket a példákat a diákok megtekinthetik, és megoldás után ellenőrizhetik, hogy helyesek–e az általuk kiszámolt eredmények, válaszok.

Az üzenetek küldése a kommunikációt segíti, amelyeket a tanár küldhet az osztályának. Ezek akár szolgálhatnak arra is, hogy a szülő is figyelemmel követhesse gyermeke tanulását. Az osztályzatok feltöltése szintén segíti a diákok és szüleik értesülését a fizika tantárggyal kapcsolatban.

Fontos szempont használhatóvá tenni az oldalt azok számára is, akiknek még viszonylag új dolog az Interneten való böngészés. A menüpontok egyértelműen utalnak azok tartalmára, így a felhasználó könnyedén megtalálhatja, amit keres. A Weblap nem csak az adott iskolába járó diákoknak adhat segítséget, mert a személyes adatokon kívül minden más bárki számára megtekinthető, aki betéved az oldalra, mivel ezekhez a részekhez nincsenek jogosultságok rendelve. Az elméleti részek mindenki számára hasznosak, míg a regisztráltak számára más esetekben is felhasználható. Például az osztályzatok és üzenetek nem csak az ütközések témakörhöz kapcsolódhatnak.

### 5. 3. Elméleti ismeretek

A Fizikaoktatás fontos szerepet tölt be, mivel sok, főleg műszaki középiskolában érettségiznek is belőle a diákok. Az ütközések témakör érdekesnek tűnhet a tanulóknak, mivel kifejezetten "ütközés"-ekre vonatkozó törvény nincs, ezért mindig más szemszögből közelíthető meg az ütközések magyarázata. Az ismereteket 2 részre lehet és ajánlott is osztani. Az alaptörvények, amik ide tartoznak, nem köthetőek kifejezetten egyik részhez sem, ezért külön vannak tárgyalva. A két rész a rugalmas illetve rugalmatlan ütközések. A törvények ismertetésekor szempont volt, hogy tagolt legyen, egymástól el legyenek különítve a különböző részek. A figyelemfelkeltés miatt a képletek ki vannak elemelve, hogy jelezzék fontosságukat. Tankönyvekben az a bevált módszer, hogy példán keresztül magyarázzák meg a fizikai folyamatot, a lendületváltozásokat valamint lendületek irányát. Ezt a módszert alkalmazva kerültek be az elméleti tananyagba a törvények mellé a kísérletek is. A feladat az, hogy röviden de mégis tömören elegendő ismeret kerüljön a diák elé a témakör megértéséhez. A tanulók közül sokan nem szeretnek több oldalas anyagot, úgymond "bebiflázni", ha ránéznek egy tömör szövegre, akkor alapból elmegy a kedvük tőle. Tehát pedagógiai szempontból fontos a tömörség, valamint a könnyen érthetőség.

### 5. 4. Szemléltetés és ellenőrzés

A korábban leírtak alapján, a fizikaoktatásban fontos szerepe van a kísérleteknek, hogy megfelelően bemutatva a diákok tudják, hogyan zajlik egy ilyen folyamat. A kísérletek bemutatása Flash animációkkal történik, amelyek lehetőségei megfelelőek az ütközések ismertetésére. Az animációk elindítása a diákra van bízva, a kísérlet előtt megnézheti az adatokat, hogy milyen testek, merre fognak mozogni, valamint a lendületek irányát és nagyságát. Egy ilyen kísérlet látható az **5. 4. a. ábrán.** 



5. 3. 4. ábra

Az animációk készítésekor szempont volt, hogy érdekesek, figyelemfelkeltőek legyenek, de ugyanakkor ne vonják el a figyelmet a lényegtől. Ne legyenek túl rikítóak, amik elterelhetik a gondolatot a fizika témától. Az egyszerűen érthetőség is hasonlóképpen fontos ezért a kísérletek egyértelműen megmutatják, hogyan történik az adott típusú ütközés.

Az ellenőrzés problémájával diákként én is szembesültem. Az otthoni megoldásra kiadott példák sok esetben hibásan kerülnek a tanár elé, mivel a diák megoldja, de nem mindig tudja megnézni, hogy a megoldásai helyesek-e, így nem is foglalkozik vele tovább. A Weblap tartalmaz példákat, amelyeket a tanár tölthet fel, az aktuális típusú ütközésekhez. Ami újításnak mondható, az hogy a példa szövege mellé a tanár megadhatja

a várt megoldásokat is. A diák a "hagyományos" módszerrel (papír, ceruza) megoldja a példát, és utána van lehetősége arra, hogy önellenőrzést végezzen a Weblap segítségével. Azért tartom ezt jó megoldásnak, mert ha például az első megoldás helyes, de a többi hibás, akkor ez motiválhatja a diákot arra, hogy megnézze még egyszer a példát, mit is ronthatott el, és ha végül kiszámolja a helyes eredményt, akkor a sikerélmény rávezetheti, hogy máskor is megoldja a házi feladatot. A példák különbözősége miatt felmerült az igény, hogy szöveges válaszok is lehetnek, ezért az adatbázis ennek megfelelően módosítva lett. A tizedes számokat szinte mindenki vesszővel írja, így a tanárnak érdemes odafigyelni, hogy a megoldásokat ő is vesszővel írja. A számításos példák mellett jelentős szerepük van, az elméleti ismeretek tudását ellenőrző feladatoknak is. Ezek a példák arra hívatottak, hogy ne csak a képletek bemagolását ellenőrizze a tanár, hanem azt is megnézze, hogy a diák érti is a tananyagot, vagy csak betanult algoritmusok alapján foglalkozik vele, konkrétabban csak a bemagolt képletek alapján számol-e?

# 6. Felhasználói dokumentáció

### 6.1. Az Appserv telepítése

Egyes iskolákban előfordulhat, hogy nem az Interneten akarják használni ezen Weblapot, hanem a saját helyi szerverükön. Ehhez az egyik megfelelő megoldás az Appserv v2.4.0. csomag, amelyben a fejlesztés is történt. A CD tartalmazza az ingyenesen is letölthető program csomagot.

A telepítés lépései:



6. 1. a. ábra

A telepítés megkezdésekor a **6. 1. a. ábrán** látható üdvözlő képernyő jelenik meg, melynek elolvasása utána 'Next' gombra kattintva lehet tovább haladni.

Choose Destination Location	n 🔀
Easy way to install for you.	Setup will install AppServ v2.4.0 in the following directory. To install to this directory, click Next. To install to a different directory, click Browse and select another directory. You can choose not to install AppServ v2.4.0, by clicking Cancel to exit Setup.
HP Mysql	Destination Directory C:\AppServ Browse Space Required: 21084 K Space Available: 394368 K
	< Back Next > Cancel

6. 1. b. ábra

A **6.1. b. ábrán** látható következő ablakban a telepítés célkönyvtárát kell megadni, a telepítő ebbe a könyvtárba fogja rakni a programcsomagot. Alapértelmezetten a C:\Appserv van megadva, de módunkban áll módosítani a 'Browse..' gomb segítségével. Majd a 'Next' gombbal továbbléphetünk.



6. 1. c. ábra

A következő **6. 1. c. ábrán** látható ablak a telepítés módjára kínál lehetőségeket. A helyi szerveren való futtatáshoz az alapértelmezett 'Typical' telepítés megfelelő. Így a 'Next' gombra kattintva máris tovább léphetünk.

Apache httpd Server			
	Server Information Please enter your server's information.		
Easy way to install for you.	Server Name (e.g. www.mydomain.com or localhost) :  localhost		
Rŝ	Administrator's Email Address (e.g. webmaster@mydomain.com) : yourname@myhost.com		
	HTTP Port (default : 80) 80		
	< Back Next > Cencel		

6. 1. d. ábra

A **6. 1. d ábra** alap beállításai is megfelelőek, a szerver elérési útját láthatjuk, az e-mail címet értelemszerűen kell megváltoztatni. Az e-mail cím megadása azért szükséges, mert a későbbiekben ide érkezhet levél, amely az Adminnak szól. Majd a 'Next' gombbal haladhatunk tovább.

E .v	Server Inform Please enter y	<b>ation</b> our MySQL infoma	ation.	MysqL
Easy way to install for you.	User Name (e.g.	apples):		
	Password (e.g. r	mypassword ) :		
	Charset (default	latin1):		
		< Back	Next>	Cencel

6. 1. e. ábra

A következő, egyben az utolsó ablakban (**6. 1. e. ábra**) érdemes megváltoztatni a felhasználó nevet (User Name) és a jelszót (Password), valamint a felírásuk is ajánlott, mert a későbbiekben ezek alapján lesz elérhető a MySQL adatbázis. A legegyszerűbb megoldást választva a felhasználó név legyen "root", a jelszó megadása nem szükséges. A 'Next' gombra kattintva elindul a tényleges telepítés.

Finished	$\overline{\mathbf{X}}$
	Setup is complete and AppServ v2.4.0 is now installed!
Easy way to install for you.	<ul> <li>✓ Start Apache</li> <li>✓ Start MySQL</li> </ul>
	Click Close to Finish Setup
	K-Black Close Cancel

6. 1. f. ábra

A **6. 1. f. ábra** a telepítés befejezését jelzi, a 'Close' gombra kattintva automatikusan elindul a szerver.

Ha a telepítés befejeződött, akkor a tálca jobb szélén az alábbi ikon jelenik meg:



A lámpa az SQL MyAdmin futását jelzi, ha zöld, akkor fut, ha sárga, akkor éppen kapcsolódik a szerverhez, ha pedig piros, akkor nem megy, ez lehet egy hiba miatt is. Az ikonnak 2 funkciója van, egyrészt információt ad, hogy fut a MyAdmin, másrészt beállítási lehetőségeket lehet vele végrehajtani.

A "lámpa" ikonra kattintva meg lehet nézni a MyAdmin beállításait, le lehet állítani, valamint a felmerülő hibákról lehet információt szerezni.

A telepítés után a helyi szerver a http://localhost, a phpMyAdmin pedig http://localhost/phpmyadmin címen érhető el.

### 6. 2. Futtatás helyi és webes szerveren

### Futtatás helyi szerveren

Az üzembe helyezéskor a következő lépések vannak még hátra:

Adatbázis létrehozása:

A CD melléklet tartalmaz két darab tömörített file-t, amelyek tartalmazzák a működéshez szükséges adatbázist. A *fizika.sql* file egy teljesen üres, míg a *fizikat.sql* file egy tesztadatokkal feltöltött adatbázist tartalmaz. Ezek a tömörített file ok, úgynevezett *dump-ok* arra szolgálnak, hogy az adatbázis univerzális legyen, tehát bármely szerverre feltölthető legyen.

Első lépésként létre kell hozni egy új adatbázist, melynek a neve lehet bármi, de meg kell jegyezni, ugyanis üzembe helyezéskor szükség lesz arra, hogy ismert legyen az adatbázis neve. Az új adatbázis létrehozását a **6. 2. a ábra** mutatja, amely a phpMyAdmin kezdőlapja.





A bekeretezett mezőben kell megadni az adatbázis nevét, jelen esetben ez legyen fizika, de továbbra is módunkban áll módosítani. Majd a "Létrehoz" gombra kattintva kész az új adatbázis.

Php. HyAdmin	Adatbázis fizika : localhost						
<b>Kezdőlap</b> fizika (-)	Adatbázis fizika megszületett.						
	SQL-kérés : [Szerkeszt] [PHP kód készítése] CREATE DATABASE `fizika` ;						
	Struktúra SQL	Export					
	Nincs tábla az adatbázisban.						
	<ul> <li>Új tábla létrehozása az adatbázisban: fizika :</li> <li>Neve :</li> <li>Mezők száma :</li> <li>Végrehajt</li> </ul>						

6. 2. b. ábra

A **6. 2. b. ábrán** baloldalt megjelenik a létrehozott adatbázis, amely jelenleg még üres. A feltöltéshez az "SQL" gombra kell kattintani.



6. 2. c. ábra

A **6. 2. c. ábrán** látható az a képernyő részlet, amelyen meg kell adni a szöveges file-t. Ez két módon lehetséges, vagy a beviteli mezőben meg kell adni az elérési utat, például A:\fizika.txt, vagy pedig a "Tallózás" gombra kattintva kell megkeresni a megfelelő file-t, amely jelen esetben legyen az üres adatbázist tartalmazó fizika.sql. A bekeretezett választó gomb arra hívatott, hogy automatikusan felismerje a kódolás módját, ezt az alaphelyzetben ("Autodetect") kell hagyni. A "Végrehajt" gombra klikkelve a **6. 2. d. ábrán** látható ablak jelenik meg, amely jelzi, hogy az adatbázisi importálás megtörtént.

Php. MyAdmin	Adatbázis fizika : localhost				
<b>Kezdőlap</b> fizika (7) ≣ admin ≣ diak	Az SQL-kérést sikeresen végrehajtott A fájl tartalmát beillesztettem. (156 Pa				
II eredm II jegyek II pelda II tanar	SQL parancs(ok) futtatása a(z) fizika adat				



Az ablak bal oldalán az importált táblák láthatóak, ez esetben 7 darabnak kell lennie.

A Weblap file-ainak feltöltése:

A CD melléklet tartalmazza weblap összes file-ját, amelyeket be kell másolni az \Appserv\www\ könyvtárba.

Az utolsó lépések:

Az adatbázis eléréséhez valamint a Weblap helyes működéséhez szükséges módosításokat a config.php fila-ban kell megtenni, az alábbi sorokban.

<pre>define("host", "");</pre>	a kiszolgáló neve (most "localhost")
<pre>define("user", "");</pre>	a felhasználónév ("root")
define("pass", "");	a jelszó ("")
<pre>define("adatb", "");</pre>	az adatbázis neve (amit a 6. 2. e ábrán megadtunk)

Az üzembe helyezési lépések végrehajtása után a weblap elérhető lesz a böngészőben, ha a következő címet írjuk be.

# http://localhost

Ilyenkor alapértelmezetten az index.htlm, tehát a kezdőlap fog megjelenni. Az Admin felület a http://localhost/admin.php, címen lesz elérhető.

#### Futtatás webes szerveren

Amennyiben a Weblapot egy Internetes szerveren akarjuk futtatni, akkor az adatbázis dump-okat tartalmazó file-okat az előbbiekben bemutatottak alapján kell importálni a webes szerverre.

Az Internetes szerveren való futtatáshoz a config.php file-ban az alábbi sorok módosítására van szükség.

define("host", "");	a kiszolgáló neve
define("user", "");	a felhasználónév
define("pass", "");	jelszó
define("adatb", "");	az adatbázis neve

Ezek a legtöbb szolgáltatónál adottak. Példaként nézzük meg a www.ultraweb.hu –ról való működéshez szükséges beállításokat.

host:	felhasználótól függő kiszolgálónév
user:	akármi lehet, amit regisztrálunk
pass:	megegyezik regisztrációs jelszóval
adatb:	megegyezik a felhasználó névvel

Befejező lépésként itt is a Weblap összes file-át fel kell tölteni az adott szerverre.

### 6. 3. A phpMyAdmin használata

Az Admin a módosításokat a phpMyAdmin -ban teheti meg, amely egy grafikus adatbázis kezelő program. Az Appserv v2.4.0.a phpMyAdmin 2.5.6 os verzióját tartalmazza. Ez a program az általános SQL utasításokat használja, de grafikus mivolta miatt jelentősen megkönnyíti a munkát.

Most nézzük meg a kész adatbázis módosításához szükséges lépéseket.





Az **6. 3. a. ábra** azt mutatja, hogyan tudjuk kiválasztani a módosítani kívánt adatbázist. A kiválasztás után a szerkesztő felületre jutunk.

Php. MyAdmin	Ad	atbázis fi	zika	: loc	alho	st				
Kezdőlap		Struktúra			SQL			Ехр	ort	
fizika (7) 💌		Tábla			Para	ancs			Sor	Tip
		admin		Q	徇	P	Ť	Ē	1	MyISA
fizika		diak		Q	徇	P	Ť	ŵ	26	MyISA
III diak		eredm		Q	徇	P	Ť	Ē	4	MyISA
III eredm		jegyek		Q,	祹	P	Ť	Ē	94	MyISA
III pelda		pelda		Q,	祹	P	Ť	Ē	4	MyISA
🔟 tanar		tanar		୍	粕	P	Ť	Ē	38	MyISA
		uzenet		Q,	祹	P	Ť	Ē	12	MyISA
		7 tábla			Össz	esen			179	
	Ê_	— Összeset	kijelöl	i / Ös	szeset	törli		A ki	jelöltekł	kel végz

### 6. 3. b. ábra

A **6. 3. b. ábrán** láthatóak az általunk kiválasztott adatbázis táblái. Itt egy kattintással az adott táblázatot láthatjuk.

Adatbázis fizika	- 1	ábla	diak	:	localhost
------------------	-----	------	------	---	-----------

:	Struktúra	Tartal	om SQL		Keresés	Beszúr	Export		Tev	ڎker	nység
								/	•		
	Mező	Tipus	Tulajdonságok	Null	Alapértelmezet	t Extra		4	Para	ancs	
	DID	int(2)		Nem		auto_incremen	nt 🗹	π	1	<u>F</u>	3
	osztaly	varchar(4)		Nem			ø	π	1	i	=
	nev	varchar(30)		Nem			ø	π	1	i	3
	szdatum	date		Nem	0000-00-00		ø	π	1	j.	=
	lakcim	varchar(30)		Nem			ø	π	1	j.	=
	jelszo	varchar(8)		Nem			ø	π_	1	j.	=
Ĺ	— Össze	set kijelöli .	/ Összeset törli	A k	újelöltekkel végze	endő művelet:	ď <b>†</b>	ĩ	<		2.

### 6. 3. c. ábra

A következő ábrán az figyelhető meg, hogy ha esetleg igény van a mezők hosszának változtatására (pl. komolyabb jelszavas védelem, több karakter kell), illetve ha egyes mezők feleslegessé válnak, akkor mi a teendő. Az egyes nyíl a szerkesztés ikonra mutat, míg a 2-es a törlésre, amire még egyszer rákérdez a program, hogy valóban törölni akarjuk-e. A bekeretezett 'Tartalom' gombbal a tényleges módosítás tehető meg, vagyis a táblába feltöltött adatokhoz férhetünk hozzá.

Mező	<b>Tipus</b> [Dokumentáció]	Hossz/Érték*	Tulajdonságok	Null	Ala
DID	INT 💌	2	~	not null 💌	
Ment * Ha a mező tip Ha backslash ** For default va	VARCHAR TINYINT TEXT DATE SMALLINT MEDIUMINT INT BIGINT FI OAT	et", akkor az ér ıfot ("") akarsz st a single value	tékeket ilyen formában írd t ezen értékek között haszná e, without backslash escap	be: 'a','b','c' ilni, használd ng or quotes, 	a b usi

### 6. 3. d. ábra

A mezőtípus módosítása a 6. 3. d. ábrán látható. Amit módosítani lehet:

a mező neve

a mező típusa

a mező hossza

A többi módosításra csak fejlesztés közben szokott szükség lenni.

1.	1	2	S0	rt by key:	Nincs	<b>~</b>	Végrehajt	
X	Τ-	V	DID	osztaly	nev	szdatum	lakcim	jelszo
	ø	t	1	11.a	Simon György	1982-06-10	2360. Gyál Mikszáth K. u 3	11
	ø	Ť	7	11.a	Kovács Jatalin	1992-05-04	6000.Kecskemét	11
	ø	Ť	13	11.b	Nagy Imre	1981-02-03	6000.Kecskemét	11
	ø	Ť	97	11.d	Kovács	1988-03-26	6000.kmét	1
	ø	Ť	98	11.a	Gizi	0000-00-00	3534	1



A 'Tartalom' gombra kattintva a **6. 3. e. ábrán** látható ablakot kapjuk. A bekeretezett rész a táblázat egy sorát, jelen esetben egy hallgató adatait tartalmazza. Az 1- es nyíl a szerkesztés, míg a 2-es nyíl a törlés gombra mutat. Ha a baloldali kockákat kipipáljuk, akkor lentebb is található egy törlés gomb, ami az összes kijelölt sorra vonatkozik. A törlés előtt itt is megerősítést kér a program.

Mező	Tipus	Funkció	Nul	Érték
DID	int(2)		~	1
osztaly	varchar(4)		~	11.a
nev	varchar(30)		~	Simon György
szdatum	date		~	1982-06-10
lakcim	varchar(30)		~	2360. Gyál Mikszáth K. u 3
jelszo	varchar(8)		~	11
⊚ Ment Vag ⊜ Besz	t y zúrás új sork	És :ént ⊂	) Vissza Vagy ) Új sor Vé	a az előző oldalra beszúrás grehajt Töröl

6. 3. f. ábra

A **6. 3. f. ábra**, a tartalom módosítására használható. A bekeretezett 'Érték' oszlopban megjelennek a felhasználó már regisztrált adatai, és ezeket kitörölve tudjuk alkalmazni a módosításokat. Ezután a 'Végrehajt' gombra kattintva megtörténik az adatbázis módosítása.

Természetesen ezek a módosítások megtehetők a többi táblán is, például ha rossz osztályzat feltöltését kell módosítani.

### 7. Tesztelés

[8]

Program illetve Weblap fejlesztés során fontos lépés a tesztelés, amelyet általában a program befejezésekor végeznek. Amennyiben a programozó befejezettnek és helyesen működőnek gondolja a programot, akkor következik a tesztelés. A tesztelés lényege, hogy minden olyan hiba kiderüljön, ami a későbbiekben hibát okozhat felhasználás közben. Tesztelés folyamán van néhány kritérium, amire érdemes figyelni:

Megvalósítja-e a program a feladat specifikációban leírtakat? Egyszerűen, értelemszerűen kezelhető, felhasználó barát-e?

### 7. 1. Üzembe helyezési teszt

A tesztelés első lépéseként azt néztem meg, hogy az ajánlott helyi szerver (Appserv v2. 4. 0.) telepítése egyszerűen és hiba nélkül elvégezhető-e, mivel az Admin, aki telepíti, lehet, hogy még nem használt helyi szervert. Az Appserv telepítését minden esetben hiba nélkül el lehetett végezni, és konfigurálás nélkül is működő képesnek bizonyult.

A második lépésben tesztelésre került, hogy a helyi szerveren való futtatáshoz szükséges lépések elvégzése után megfelelően működik–e a Weblap. A file-ok ...\Appserv\www\ könyvtárba való bemásolása után a helyi szerveren az elvártaknak megfelelően lehetett elérni az Internet Explorer böngészőben. Amennyiben az iskolának van helyi szerver gépe, akkor elég azon telepíteni.

Tesztelésre került az is, hogy Internete szerverre is megfelelően lehet-e telepíteni az oktató honlapot. A tesztelésre a www.ultraweb.hu ingyenes oldalt használtam, amely megfelelő volt, mivel itt támogatottak a php-s oldalak, valamint MySQL adatbázis is használható, arról nem is szólva, hogy phpMyAdmin-t is tartalmaz. Az üzembe helyezés után elérhető és működőképes volt a weblap, amely az alábbi címen meg is tekinthető: http://fizika2005.uw.hu

### 7. 2. Funkcionális tesztelés

A Weblapot működés közben teszteltem, az elsődleges szempont az volt, hogy a feladat specifikációban leírt funkciókat megvalósítja-e. A másodlagos figyelmet arra fordítottam, hogy a menüpontok a nevüknek megfelelő funkciót hajtsák végre, valamint

hogy ezek a feladatok helyesen mennek-e végbe. Lépésről lépésre ki lett próbálva minden menüpont, majd az adatbázisban ellenőriztem, hogy az adatok ténylegesen fel lettek-e töltve. A tesztelésnél jelentős szerepet kapott a beléptetés és a nyomon követés, mivel a Weblap személyes információkat is tartalmaz, amelyeket illetéktelen személyek nem láthatnak.

Eredményként azt kaptam, hogy a célkitűzésben leírt funkciók helyesen működnek, az adatbázisban jól jelennek meg az adatok és a beléptetés is az elvártaknak megfelelően működik.

# 8. Összegzés

A szakdolgozat célja egy olyan oktató honlap volt, amely középiskolákban támogatja a fizikaoktatást. A Weblap segítséget nyújt az elméleti ismertekben, valamint az adatbázisnak köszönhetően adatokat is tárolni lehet a diákokról, a tanárokról, valamint az önellenőrzés miatt példákat is tartalmaz. A felhasználhatóság is fontos volt, mivel sokaknak még újdonság az Interneten való böngészés, ezért mindenki számára kezelhetővé kellett tenni. A kommunikáció – üzenetküldés – is cél volt, mivel ez még kevés iskola honlapján fordul elő. A beléptető rendszerre a korlátozott hozzáférés miatt volt igény.

Úgy gondolom, hogy a kitűzött célt sikerült elérni, mivel egy olyan Weblap lett a fejlesztés eredménye, amely megoldja a tervezett feladatokat. Nem minden megoldás van még webes felületen megoldva, de a hiányzó űrlapokat a phpMyAdminban meg lehet valósítani. Későbbi fejlesztési cél, hogy ezek az adatbázis műveletek a webes felületen is megvalósuljanak.

A tervezés folyamán cél volt az is, hogy az oktatási szempontok is figyelembe legyenek véve, lévén, hogy oktató Weblapról van szó. Az elméleti ismereteket a kitűzés szerint sikerült rövidre, tömörre, de mégis érthetőre alakítani, valamint a szimulációk megfelelően visszaadják a kívánt ütközés lényegét. A hozzáférési jogoknak köszönhetően a Tanár – diák közötti kommunikáció megoldott lett, jogosulatlan személy nem férhet személyes információhoz.

Fejlesztési lehetőség is adott a végső Weblapból, mivel ez egy alap lehet, egy tovább bővített oldalhoz. Az adatbázist módosítva akár több tantárgy is szerepelhet, de adott a lehetőség arra is, hogy a Fizika témakört bővítsük ki. A választott objektum orientáltság miatt, a további osztályok és függvények könnyen elkészíthetők.

# 9. Irodalomjegyzék

[1]	Balogh Lászlóné dr.:
	Fizika I. szakközépiskoláknak
	Műszaki Könyvkiadó, 2002
[2]	Csernovszky Zoltán
	Tömegközéppont, lendület, ütközések
	http://www.sulinet.hu/tovabbtan/felveteli/2001/3het/fizika/fizika3.html
	2005. április
[3]	Mazik István:
	UML alapú rendszermodellezés
	BME méréstechnikai és Információs Rendszerek tanszék, 2004
	(http://www.inf.mit.bme.hu/~varro/uml/slides/uml.ppt)
	2005. március
[4]	Bércesné Novák Ágnes
	Adatmodellezés, Egyed- Kapcsolat modell
	http://www.banki.hu/jegyzetek/adatbazis/manager/modellek.pdf
	2005. április
[5]	Nagy Gusztáv
	A WSDM weboldal- tervezési módszer a gyakorlatban
	http://www.phpconf.hu/2005/media/eloadasok/WSDM.pdf
	2005. április
[6]	Julie C. Meloni:
	A PHP, a MySQL és az Apache használata
	Panem Kiadó, Budapest, 2004
[7]	STS
	Felhasználok beléptetése és követése
	http://www.prog.hu/cikkek/425/Felhasznalok+beleptetese+es+kovetese+1.html
	2005. március
[8]	Dr. Balla Katalin
	Szoftver minőség és menedzsment: 15. Tesztelési technikák
	http://www.iit.bme.hu/~balla/SzMM15-teszteles1.ppt
	2005. május