

ZÁRÓDOLGOZAT

Név: Kiss Gábor Osztály: 13.D

Bookstore Management System

Tartalom

1. Bevezetés
 - 1.1 Záródolgozat téma kiválasztása
 - 1.2 Alkalmazás tervezése
 - 1.3 Programnyelv kiválasztása, előkészületek
 - 1.4 Adatbázis
 - 1.5 Köszönetnyilvánítás
 2. Felhasználói dokumentáció
 - 2.1 Rendszer Követelmény
 - 2.2 Alkalmazás telepítése
 - 2.3 Alkalmazás használata
 - 2.4 Gyakran ismételt kérdések
 - 2.5 A program készítőjének elérhetősége
 3. Fejlesztői dokumentáció
 - 3.1 Adatbázis
 - 3.1.1 Account tábla
 - 3.1.2 Hozzászólás tábla
 - 3.1.3 Mentetttopic tábla
 - 3.1.4 Topic tábla
 - 3.2 Forrás kód
 - 3.3 Java – PHP kapcsolat
 - 3.4 PHP kódok
 - 3.5 Főbb változók
 - 3.6 Admin felület
 - 3.7 Fejlesztési lehetőségek
 - 3.8 Milyen nehézségekkel találkoztam?
 - 3.9 Tesztdokumentáció
 4. Irodalomjegyzék
-

1. Bevezetés

1.1 Záródolgozat téma kiválasztása

A Bookstore Management System egy webalapú alkalmazás, amely megkönnyíti a könyvkereskedést különféle kategóriákban, mint például életrajz, programozás és menedzsment. A projekt célja az adminisztráció papírmunkájának csökkentése és a vásárlási folyamatok digitalizálása. A téma választásának indoka, hogy a könyváruházak számára egyszerűbb, központosított adatbázis-alapú megoldást nyújtson a készletek és ügyféladatok kezelésére.

1.2 Alkalmazás tervezése

Miután eldöntöttem, hogy a könyvesbolt kezelő rendszer tervezésére fókuszálok, kezdeti látványterveket készítettem az alkalmazás számára. Az első lépésként egy alapvető tervet dolgoztam ki, amely a felhasználói élmény javítására és a funkciók könnyebb elérésére összpontosított.



Felhasználói használati eset diagram: 

1.3 Programnyelv kiválasztása, előkészületek

Miután eldöntöttem, hogy alkalmazást szeretnék létrehozni, akkor kerültem szembe azzal a problémával, hogy rengetek platform van egy alkalmazás létrehozására. Időbe telt, mire átnéztem párat ezek közül, de végül a 'Visual Studio Code' nyerte el a tetszésem, amelyben HTML, CSS, Java, Bootstrap-et használtam

1.4 Adatbázis

Az adatbázis létrehozására, illetve annak szerkesztéséhez az XAMPP programot használtam, ami teljesen megfelelt mindenre, amire szükségem lehetett.

Természetesen az adatbázissal kapcsolatban is rendelkeztem egy tervvel.

A rendszerben használt különféle táblázatok a következők:

1. Admin
2. Book
3. Category
4. Contact
5. Register
6. Order

1. Table Name: Admin

- **Primary Key:** a_id
- **Description:** For storing Admin login details.

Field	Type	Description
a_id	int(4)	Used to store Admin ID
a_unm	varchar(3)	Used to store Username of Admin
a_pwd	varchar(30)	Used to store Password of Admin

2. Table Name: Book

- **Primary Key:** b_id
- **Description:** Stores Book details.

Field	Type	Description
b_id	int(10)	Used to store Book ID
b_nm	varchar(50)	Used to store Book Name
b_cat	int(6)	Used to select or store the Book ID of different Categories
b_desc	longtext	Used to store the Description of the Book
b_price	int(4)	Used to store the Price of Book
b_img	varchar(50)	Used to store the Image Name of Book
b_time	int(20)	Used to store the Time of Inserted book

3. Table Name: Category

- **Primary Key:** cat_id
- **Description:** Stores Category Names.

Field	Type	Description
cat_id	int(10)	Used to store the Category ID
cat_nm	varchar(50)	Used to store the Category Name

4. Table Name: ContactS

- **Primary Key:** c_id
- **Description:** Stores details for Contact Us.

Field	Type	Description
c_id	int(4)	Store Contact ID of Client/User
c_fnm	varchar(100)	Store Full Name of User
c_mno	int(10)	Store Mobile Number of Client/User
c_email	varchar(60)	Store the E-Mail Address of Client/User

Field	Type	Description
c_msg	longtext	Store The Message or Query of The Client/User
c_time	varchar(20)	Store The Time of Contact Page inserted The Data

5. Table Name: Register

- **Primary Key:** r_id
- **Description:** Details for Registration Visitors or Users.

Field	Type	Description
r_id	int(8)	Store User Registration ID
r_fnm	varchar(100)	Store Full Name of User
r_unm	varchar(50)	Store Username of User
r_pwd	varchar(30)	Store the Password of User
r_cno	varchar(10)	Store the Contact Number of User
r_email	varchar(60)	Store E-Mail Address of User
r_time	varchar(20)	Store Time of Registration of User

6. Table Name: Order

- **Primary Key:** o_id
- **Description:** Details for Order.

Field	Type	Description
o_id	int(11)	Store Order ID
o_name	varchar(30)	Store Full Name of User
o_address	varchar(200)	Store address of User
o_pincode	int(20)	Store city Pincode
o_city	varchar(30)	Store the City Name
o_state	varchar(30)	Store State
o_mobile	bigint(20)	Store Mobile Number
o_rid	int(8)	Store Register ID

1.5 Köszönetnyilvánítás

Annak ellenére, hogy ebben az évben igencsak keveset tartózkodtunk az iskolában, rengeteg segítséget és támogatást kaptam Kovács László tanár úrtól, gyakorlati és elméleti szinten is.

2. Felhasználói dokumentáció

2.1 Rendszer Követelmény

Hardverkövetelmény

- Rendszertípus 32 bites operációs rendszer.
- Windows 7/8/8.1/10
- Linux Ubuntu / Light ubuntu
- Mac OS
- 350 MB RAM

Szoftverkövetelmény

- Wamp szerver
- MySQL
- Böngésző
- PHPMyAdmin

Klienstoldali Eszközök

- Processzor: PC Dual-core processzorral vagy annál gyorsabb ajánlott: 2.20 GHz processzor.
- RAM: 512 MB vagy nagyobb ajánlott.
- Merevlemez: 45 MB szabad hely szükséges a rendszermeghajtón, vagy több.
- Operációs rendszer: Windows vagy nyílt forráskódú 32/64 bites operációs rendszer, vagy újabb verziók.
Böngészők: Mozilla Firefox 2.0 / Internet Explorer 8.0 vagy újabb / Google Chrome.

2.3 Alkalmazás használata

2.4 Gyakran ismételt kérdések

- -Elérhető lesz különféle webáruházakban?
- *Igen amint a szükséges dokumentumokat kitöltöttem.
- -Mennyi különféle felhasználót hozhatok létre?
- *Bármennyit. Nincs megkötve viszont nem árt figyelem előtt tartani, hogy 6 hónap inaktivitás után töröljük a nemhasznált felhasználókat!
- -Abban az esetben, ha problémám van a programmal hol jelezhetem ezt?
- *Amennyiben a fórumon ezt nem teheti meg abban az esetben a 'elérhetőség' részénél
- feltüntetett email címen lehet jelezni ezt

2.5 A program készítőjének elérhetősége

- E-mail cím:kissgabor5622@gmail.com
- Telefonszám: +36304559752

3. Fejlesztői dokumentáció

3.1 Adatbázis tervezése

A tervezési fázis célja, hogy megoldást találjunk a követelmények által meghatározott problémára. A rendszertervezés célja a rendszer moduljainak azonosítása, ezen modulok specifikációjának meghatározása, és annak megértése, hogyan lépnek interakcióba egymással a kívánt eredmény elérése érdekében. A tervezési folyamat célja egy olyan modell elkészítése, amelyet később a rendszer megépítéséhez használhatunk. Ezt a modellt nevezzük a rendszer tervének.

A rendszertervezés folyamata a rendszer architektúrájának, komponenseinek, moduljainak, interfészeinek és adatainak meghatározása annak érdekében, hogy megfeleljen a meghatározott követelményeknek.

Általában a tervezés két szakaszban történik:

- Fizikai tervezés
- Adatbázis-tervezés

Fizikai tervezés

A fizikai tervezés egy grafikus ábrázolása a rendszernek, bemutatva a rendszer belső és külső entitásait, valamint az adatáramlást ezek között az entitások között. A belső entitás olyan entitás, amely a rendszeren belül adatokat alakít át. A fizikai tervezést olyan diagramokkal ábrázoljuk, mint például az adatáramlási diagramok (DFD), entitás-kapcsolat (E-R) diagramok, és használati eset (Use Case) diagramok.

1. Adatáramlási diagram (DFD)

Az adatáramlási diagramok (DFD) grafikus ábrázolást nyújtanak az információs rendszeren keresztül történő adatáramlásról. Az ilyen diagramokat a rendszer elemzői használják az információfeldolgozó rendszerek tervezésére, de akár egy egész szervezet modellezésére is alkalmasak. A DFD fő előnye, hogy átfogó képet ad arról, milyen adatokat dolgoz fel a rendszer, milyen átalakításokat hajt végre, milyen adatokat tárol, és hová áramlanak az eredmények.

DFD-ben használt szimbólumok:

- **Adatáramlás:** A folyamatokat köti össze, a nyílhegy mutatja az adat áramlásának irányát.
- **Folyamat:** Az adatot átalakító műveletet hajt végre.
- **Bemenet/Kimenet:** Az adatok bemenetére vagy kimenetére szolgál.

DFD szintek:

- **0. szintű DFD:** Egy weboldal alapvető adatáramlását mutatja.
- **1. szintű DFD:** Részletesebb adatáramlást mutat az előzőhöz képest.

Egyéb diagramok:

- **Folyamatábra:** A rendszer folyamatainak lépéseit mutatja be.
- **Felhasználói folyamatábra:** A felhasználó által végrehajtott lépések vizualizációja.

3.6 Admin felület

Az Admin funkciói:

- Ez a modul főként a következő feladatokat tartalmazza:
- Kategória bejegyzése.
- Kategória lista.
- Új könyv hozzáadása.
- Könyv megtekintése.
- Az ügyfél által küldött üzenet megtekintése.

13. Admin Login Page (New Template) ☐

14. Admin Home Page ☐

15. Add Category (Admin) ☐

3.7 Fejlesztési lehetőségek

Súgó modul

A modul használatával a felhasználók segítséget kaphatnak a rendszer eléréséhez. Ebben a modulban a rendszer összes funkcióját ismertetjük. A felhasználó pedig könnyedén elérheti a teljes modult ezzel a funkcióval.

Online fizetési modul

A felhasználó ezen funkció segítségével online fizethet. A jövőben online fizetést is bevezetünk, hogy a fizetést megkönnyítsük a felhasználó számára.

Többnyelvű

Ebben a rendszerben hozzáadjuk a többnyelvűt, így a felhasználók különböző nyelveken dolgozhatnak és könnyen megérthetik.

3.8 Milyen nehézségekkel találkoztam?

Valóban sok kihívással találkoztam. Elsősorban az iskola mellett nehéz volt belekezdeni egy új programozási nyelv tanulmányozásába, ami a kódomon is látszik, hiszen sok helyen kommentek segítik a megértést. Emiatt a belépési menü is eléggé zavaros lett.

Miután megismertem a lehetőségeimet, újabb akadályba ütköztem: nem tudtam összekapcsolni az alkalmazást az adatbázissal. Bár az interneten sokféle megoldás volt, mindegyik eltérő, mivel a probléma nem mindenkinél ugyanaz.

Kezdetben nem is gondoltam arra, hogy PHP-t fogok használni az alkalmazás összekapcsolására. A kutatásaim végül arra vezettek, hogy ez a legkényelmesebb megoldás számomra.

3.9 Tesztdokumentáció

Fekete doboz tesztelés

A fekete doboz tesztelés egy olyan szoftvertesztelési módszer, amely az alkalmazás funkcionalitását a specifikációk alapján vizsgálja. Más néven specifikáció-alapú tesztelésnek is nevezik. Ezt a típusú tesztelést független tesztelő csapatok végzik a szoftvertesztelési életciklus során.

Ez a tesztelési módszer alkalmazható minden szoftvertesztelési szinten, például egységtesztelés, integrációs tesztelés, rendszer- és elfogadási tesztelés során.



Ez a módszer a neve szerint úgy van elnevezve, mert a szoftverprogram a tesztelő szemében olyan, mint egy fekete doboz; azaz nem látható belülről. Ez a módszer a következő kategóriákban próbál hibákat találni:

- Hibás vagy hiányzó funkciók
- Interfész hibák
- Hibák az adatstruktúrákban vagy külső adatbázis hozzáféréseinél
- Viselkedési vagy teljesítmény hibák
- Inicializálási és lezárási hibák

Előnyök

- A tesztelés a felhasználói szempontból történik, és segít feltárni a specifikációkban található eltéréseket.
- A tesztelőnek nem szükséges tudnia programozási nyelveket vagy azt, hogyan lett a szoftver megvalósítva.

Fehér doboz tesztelés

A fehér doboz tesztelés egy olyan tesztelési technika, amely a program struktúráját vizsgálja, és tesztadatokat derivál a program logikájából/kódjából. Más néven üveg doboz tesztelésnek, tiszta doboz tesztelésnek, nyitott doboz tesztelésnek, logika-vezérelt tesztelésnek, vagy struktúra tesztelésnek is nevezik.

A fehér doboz tesztelés során a kód struktúráját vizsgáljuk. Ha ismerjük a termék belső struktúráját, a tesztek biztosítják, hogy a belső műveletek a specifikációknak megfelelően történjenek, és minden belső komponens megfelelően legyen tesztelve.

Fehér doboz tesztelési technikák

- **Nyilatkozat fedezet:** Ez a technika arra irányul, hogy minden programozási nyilatkozatot teszteljen minimális tesztekkel.
- **Ág fedezet:** Ez a technika egy sor teszt futtatását jelenti, hogy biztosítsuk, hogy minden ágat legalább egyszer teszteltek.
- **Útvonal fedezet:** Ez a technika az összes lehetséges útvonal tesztelésére vonatkozik, ami azt jelenti, hogy minden nyilatkozat és ág le van fedve.

Előnyök

1. Kényszeríti a teszt fejlesztőt, hogy alaposan megfontolja a megvalósítást.
2. Feltárja a „rejtett” kód hibáit.
3. Támogatja a kódot vagy más problémákat a legjobb programozási gyakorlatokkal kapcsolatban.

Szürke doboz tesztelés

A szürke doboz tesztelés egy olyan tesztelési technika, amely korlátozott információval rendelkezik a rendszer belső működéséről. A szürke doboz tesztelők hozzáférnek a követelmények részletes tervezési információihoz.

A szürke doboz tesztelés a célrendszer állapot-alapú modellek, UML diagramok vagy egyéb dokumentációk alapján történik. A szürke doboz tesztelés technika a szoftvertermék vagy alkalmazás tesztelésére vonatkozik részleges ismeretekkel az alkalmazás belső működéséről.

Admin Bejelentkezési Részletek

- Felhasználónév: Admin
- Jelszó: Admin

Várt Eredmény:

- Ha a mezők üresek, hibaüzenet jelenik meg a mezők kitöltésére vonatkozóan.
- Ha a jelszó vagy felhasználónév nem létezik, hibaüzenet jelenik meg a helyes adatokkal kapcsolatban.

Bejelentkezési Részletek

- Felhasználónév: kiss
- Jelszó: kiss

Várt Eredmény:

- Ha a mezők üresek, hibaüzenet jelenik meg a mezők kitöltésére vonatkozóan.
- Ha a jelszó vagy felhasználónév nem létezik, hibaüzenet jelenik meg a helyes adatokkal kapcsolatban.

Regisztrációs Részletek

- Felhasználónév:
- Jelszó:
- Teljes Név:
- Biztonsági Kérdés Válasz:

Várt Eredmény:

- Ha a mezők üresek, hibaüzenet jelenik meg a mezők kitöltésére vonatkozóan.
- Ha a jelszó vagy felhasználónév nem létezik, hibaüzenet jelenik meg a helyes adatokkal kapcsolatban.
- Ha a jelszó kevesebb, mint 8 karakter, hibaüzenet jelenik meg.

Megrendelési Részletek

- Teljes Név:

- Cím:
- Kapcsolattartó Szám: ÜRES

Várt Eredmény:

- Ha a mezők üresek, hibaüzenet jelenik meg a mezők kitöltésére vonatkozóan.
- Ha a kapcsolattartó szám nem numerikus, hibaüzenet jelenik meg.

4 Irodalomjegyzék

- Szín választás : https://www.w3schools.com/colors/colors_picker.asp
- PHP parancsok : <https://www.php.net/manual/en/index.php>
- Java segítség : <https://www.w3schools.com/java/>
- Java olvasmány : Alkalmazásfejlesztés Android Studio rendszerben