



ÓBUDAI EGYETEM

Alba Regia Műszaki Kar

Mérnöki Intézet

SZAKDOLGOZAT

On-line időpontfoglaló web alkalmazás fejlesztése



OE-AMK

2018

Hallgató neve:

Hallgató törzskönyvi száma:

Szöböllődi Márk

T000401/FI12904/N-S



ÓBUDAI EGYETEM

Alba Regia Műszaki Kar
Mérnöki Intézet

Sorszám: AMK-1694/2018

SZAKDOLGOZAT FELADATLAP

Hallgató neve: Szöböllődi Márk

Törzskönyvi száma: T000401/FI12904/N-S

Neptun kódja: [REDACTED]

Képzés: Mérnök informatikus alapszak – Vállalati információs rendszerek specializáció

A dolgozat címe: On-line időpontfoglaló web alkalmazás fejlesztése

A dolgozat címe angolul: On-line Appointment-Fixing Web Application Development

Feladat: Klímaszerelő vállalkozás részére új weboldal tervezése és fejlesztése, melynek megjelenése reszponzív. Az alkalmazás lényegi részét képezi egy on-line időpontfoglaló rendszer tervezése, fejlesztése. Az alkalmazás legyen képes eltárolni a foglalásokat a megfelelő jogosultságok biztosításával.

A hallgató tervezze meg az elkészülő alkalmazást és az adatok tárolásához használni kívánt adatbázist! A szakdolgozat tartalmazza a fejlesztés során használt reszponzív technológiák leírását az elkészített szoftver tervét és részletes dokumentációját.

Intézményi konzulens neve: dr. Nagyné Dr. Hajnal Éva egyetemi docens

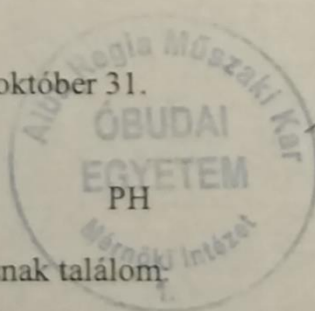
A kiadott téma elévülési határideje: 2020.02.10.

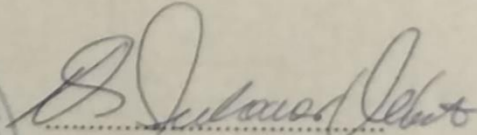
Beadási határidő: 2018.12.15.

A záróvizsga tárgyai: Számítógép architektúrák, Vállalati információs rendszerek

A szakdolgozat nem titkosított.

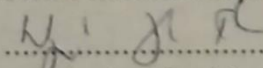
Kiadva: Székesfehérvár, 2018. október 31.




Dr. Seebauer Márta
intézetigazgató

A dolgozatot beadásra alkalmasnak találom.

20.18. december hó.....nap


.....
intézményi konzulens



HALLGATÓI NYILATKOZAT

Alulírott szigorló hallgató kijelentem, hogy a szakdolgozat/diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem.

Egyéb jelentős segítséget nem vettem igénybe.

Az elkészült szakdolgozatban/diplomamunkában található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Kelt: 2018. december 13.

Székelyi Erik
hallgató aláírása

Tartalomjegyzék

1	Bevezetés	6
1.1	Feladat meghatározása	7
1.2	Célok	8
2	Felhasznált szoftverek	9
2.1	Visual Studio Code	9
2.2	Xampp	10
2.3	Ezit Webtárhely	10
2.3.1	cPanel	11
2.4	Operációs rendszer	11
3	Fejlesztői dokumentáció	11
3.1	Rendszer komponensei	12
3.2	Adatbázis terv	13
3.2.1	Adatbázis táblák	13
3.2.2	Relációs adatmodell	15
3.3	A rendszer felépítése	15
3.3.1	Felhasználói felületek	16
3.3.2	Modulok működése	21
3.3.3	Biztonságtechnikai szempontok	37
4	Tesztelés	38

5 Összegzés	41
6 Summary.....	42
7 Irodalomjegyzék	43
8 Ábrajegyzék.....	44

1 Bevezetés

Apukám születésem óta foglalkozik klímaberendezések értékesítésével, szerelésével és különböző hűtési/fűtési megoldások kivitelezésével. Nemrégiben alapított egy külön céget, a QualityWorld Kft.-t azzal a céllal, hogy terjeszkedjen, és ehhez szeretnék hozzájárulni egy bemutatkozó weboldallal, amelynek segítségével az ország még több pontján megismerhetik a vállalatot és tevékenységi körét. Azért döntöttünk saját fejlesztés mellett, mivel jelentős költségekkel járna egy kész szoftver megvásárlása, esetleges szolgáltatásként való bérlete, illetve még utána is sok időt, energiát és pénzt felemésztene a testreszabása, átalakítása. Ami még a döntésünk mellett érvelt, az a funkciók tárháza. Nem szívesen adnánk ki több pénzt feleslegesen, mert nem tudnánk kihasználni a funkciókat. Kisebb cég lévén az eddig papíron, szétszórta táblázatokban vezetett adatokat és folyamatokat szeretnénk egységesíteni, átláthatóbbá, kezelhetőbbé tenni.

A vállalkozás első sorban magánszemélyek részére árusít többféle klímaberendezést, tervez szellőztető rendszert, hőcserélő rendszert, egyéb hűtési/fűtési megoldásokat kínál és ezek beépítésében közreműködik. Mindemellett cégek részére is rendelkezik ipari kivitelű berendezésekkel, megoldásokkal, rendszerekkel és ezekhez megfelelő szakértelemmel, szervízháttérrel és rendelkezésre állással.

Elsődleges feladat egy olyan felület kialakítása, ahol a weboldal látogatói előzetes regisztráció után képesek időpontot foglalni a cég által nyújtott szolgáltatások egyikére, a vállalat által megadott szabad időpontok közül. Az időpontkérésekről értesítést kap a vállalat adminisztrátora és visszaigazolásukról a megrendelő is e-mail formájában. Mivel a megrendelők személyes adatai a szoftver MySQL [1] adatbázisában lesznek tárolva és még sok más érzékeny adat, ezért ügyelnünk kell, hogy ne kerülhessenek illetéktelen kezekbe, biztonságosan tudjuk tárolni őket. A weboldal megjelenésével sugallja az általuk nyújtott szolgáltatások minőségét. Reklám funkcióját is ellátja a weboldal, így nagy hangsúlyt fektettünk az online előre sorolás, keresőoptimalizálás és célzott hirdetésekre is, hogy minél több helyre eljusson a cég híre ezzel megelőzve a konkurens vállalatokat.

Fontosnak tartom, hogy egy vállalatnak legyen egy megfelelő weboldala, ahol a kínált szolgáltatásokról elegendő információt nyújt az arra érdeklődőknek és bizalmat szavazzon további együttműködés fejében. Az emberek döntő többsége manapság bármit is szeretne

vásárolni, megvalósítani legelőször az interneten keres rá és ha véletlen nem talál a számára megfelelő szolgáltatást nyújtó megoldást, utána fordulhat hagyományos fórumokhoz, például újsághirdetések vagy még ez előtt kompromisszumot köt egy első között talált online weboldal nyújtotta szolgáltatásokkal.

A weboldal kinézetén és rajta tárolt információ minőségén és mennyiségén felül fontos a megjelenő menüpontok logikája, az oldal sebessége, rendelkezésre állása és a megjelentett funkciók hibátlan, megbízható működése. Ezek után, ha a weboldal már a legelején elnyeri az érdeklődő tetszését és még a keresett szolgáltatást is megtalálta, onnantól már csak pár kattintás a megrendelés véglegesítése.

1.1 Feladat meghatározása

A feladat tehát egy weboldal készítése, mely először is egy bemutatkozó főoldalból áll. Itt megjelenik minden főbb információ a cégről, múltjáról és tevékenységeiről. A főoldalon megjelenő elérhetőségeken felül lehetőség nyílik felvenni a kapcsolatot a céggel regisztráció nélkül egy formon keresztül név, e-mail cím és az üzenet megadásával.

Az ajánlatkérő felületre navigálva először egy bejelentkező ablakba futunk. Itt e-mail cím, jelszó párossal lehet belépni. Ha még nincs felhasználói fiókunk, akkor lehetőség nyílik egy regisztrációs form kitöltésére, majd e-mail visszaigazolásban érkező aktiválás után be lehet jelentkezni. Regisztrált felhasználó esetén lehetőség nyílik egyből a bejelentkezésre. Sikeres bejelentkezés után az időpontfoglaló modul fogad minket, ahol az aktuális heti beosztást látjuk, mind a szabad és mind a foglalt időpontokat külön jelölve. Szabad időpontra kattintva előugrik egy foglalási ablak, ahol további adatokkal tudjuk a foglalás részleteit pontosítani. Amennyiben a cég egyik alkalmazottja, adminisztrátora jelentkezik be (admin) még egyéb feltöltési és állítási lehetőségek is megjelennek. Például szabad időpontok felviteli lehetősége, meglévő foglalások módosítása, szerelők feltöltése az adatbázisba, szerelők hozzárendelése a foglalt időpontokhoz. Minden felhasználó látja továbbá az általa lefoglalt időpontokat, az admin felhasználók pedig az összes lefoglalt időpontot, azok részletes adatait és módosítási jogosultsággal bír felettük.

1.2 Célok

Olyan weboldal fejlesztése, ahol

- ismertetésre kerül a cég tevékenységi köre egy modern, letisztult és ízléses bemutatkozó főoldalon, ami színeiben is illik a klímatiszáláshoz.
- az oldal megjelenése reszpónzív [3], tehát nem csak asztali számítógépen vagy laptopon, hanem okostelefonon és tableten is kezelhető marad a tartalom, emellett esztétikus az elrendezés, nem csúszik félre a tartalom és esetleges átfedésektől mentesen működik.
- előzetes regisztráció után az oldal látogatói képesek szolgáltatásokra időpontot foglalni minden részlet ismertetésével anélkül, hogy kérdés maradjon bennük és ezeket tudják a jövőben módosítani, lemondani.
- a cég munkatársai, az oldal adminisztrátorai könnyen áttekinthetik a megrendeléseket, az ott szereplő adatok alapján felvehetik a kapcsolatot a megrendelőkkal, pontosan informálhatják a dolgozóikat a munka részleteivel kapcsolatban, pontosabb előzetes kalkulációval szolgálhatnak a cég vezetői felé.
- az adminisztrátorok további módosításokat is végezhetnek a megrendelésekkel kapcsolatban.
- a hibásan foglalt időpontokat a cég adminisztrátorai felszabadíthatják, a még szabad időpontokat törölhetik, illetve a foglalásban szereplő megrendelés teljesítése után lezártnak könyvelhetik.

Mindezt biztonságos környezetben tehetik, ahol az ügyfelek és a cég adatai is megfelelően vannak tárolva, titkosítva és védve az esetleges rosszindulatú támadások, behatolások ellen. A biztonságtechnikai szempontokról bővebben még a következő pontokban szó esik. A weboldalt a jövőben a megrendelői visszajelzések alapján bővíteni/fejleszteni szeretnénk, ezért a legfontosabb szempont, hogy technológiai oldalról is modern, korszerű alapokon álljon. Álljon mögötte egy átlátható, optimalizált adatbázis és a legtöbb számítást és műveletet szerver oldalon végezze el, ezzel is növelve a kiszolgálás sebességét.

2 Felhasznált szoftverek

Ebben a pontban ismertetem a weboldal fejlesztéséhez, majd üzemeltetéséhez szükséges rendszereket. Saját elképzelésemet, ami alapján ezeket az alkalmazásokat és szolgáltatásokat választottam. Majd röviden bemutatam őket és a végleges konfiguráció felépítését. Részletes beállításait is ismertetem.

2.1 Visual Studio Code

Az alkalmazás fejlesztését a fent megjelölt Visual Studio Code-ban hajtottam végre, mely ingyenes otthoni és kereskedelmi használatra is. Nekem ez tűnt a legkényelmesebbnek programozásra, gyorsan gépelhetőek benne a fejlesztett alkalmazás kiterjesztésének megfelelő tag-ek, függvények, változók. Jól tagol, jól elkülönülő színekkel jelöl, egyszerű felülete van, ami még témázható is. Utóbbi részben fontos a szemkímélés céljából, kevésbé fárasztja a szemet a sötétebb téma délután, mégis jól olvasható, illetve tűző napsütés esetén egy világosabb, fehér háttérű témával jobban elkülönülnek a karakterek, kontrasztosabbak, ezáltal jobban olvashatóvá válik nappal. Rengeteg kiegészítő, modul letölthető hozzá, mely például a tesztelést, szerver hosztolást teszi lehetővé és a jövőben hasznos lehet csapatban való munkákhoz is mivel van benne GitHub integráció. Korábban próbáltam Adobe Dreamweavert, Notepad++-t, Sublime_Text-et és Atom-ot is. Legutóbbi nagyon hasonló az általam választott Visual Studio Code-hoz, de kevésbé tűnt megbízhatónak, néhol bugos volt a bővítmények működése terén. Továbbá támogatja a több ablakos módot. Szélesebb monitor esetén kényelmesen elfér egymás mellett a Project Explorer, egy html oldal és hozzá tartozó stílus css, esetleg javascript vagy php programkód is. Összességében minden kényelmi funkcióval rendelkezik, programozási nyelvtől függetlenül használhatom a megszokott szoftveremet, minden gyorsgombot elég egy környezethez megtanulni, nincs tele színes ikonokkal, amik elvonhatnák a figyelmet a fejlesztésről, tényleg csak annyit tartalmaz, amennyi fontos. Legfrissebb, 1.29.1 verziójú állt rendelkezésemre fejlesztésem során.

2.2 Xampp

Eleinte saját helyi hálózaton hosztolt szerverre (Localhost) fejlesztettem, teszteltem. Az Xampp platformfüggetlen, szabad felhasználású, nyílt forráskódú webservert csomag. Enélkül nem tudtam volna PHP kódot futtatni és MySQL-ben dolgozni. Egyszerű telepítés és rövid konfigurálás után már rendelkezésre is állt. MySQL szervert kellett még utána beállítani a phpMyAdmin oldalon. Itt az adatbázis felhasználót kellett létrehozni minden jogosultságával (esetleg többet), illetve utána az adatbázist, azon belül táblákat a megfelelő mezőkkel és jöhetett a feltöltés a tesztadatokkal. Első tesztelések után hamar váltottam a végleges megoldásra, saját bérelt webservert, amikor már biztonságosan elkülönítve tudtam tartani az adatbázisomat.

2.3 Ezit Webtárhely

Később béreltem saját SSD webtárhelyet. Ajánlások alapján jutottam ez az Ezit Kft.-hez. Náluk teljesen el tudtam intézni a weboldalhoz tartozó összes ügyet. Saját ügyfélkapu regisztráció után itt regisztráltam a domain címet kedvezményes akcióban több évre. A címet saját szerverükön elhelyezett tárhelyemre irányították azonnal. Az ügyfélkapunk keresztül minden nyomtatványt le/fel tudtam tölteni, amely a domain tulajdonlásához elengedhetetlen volt. Az ügyfélkapun keresztül továbbá nyomon követhetem a megrendelt szolgáltatásaimat, azoknak az esedékes díjait. Bármikor bővíthetem, esetleg rendelhetek VPS szervert, de ez jelen esetben szükségtelen. Cégek esetén kényelmes, hogy az Áfás számlák is bármikor letölthetőek. Van saját Wiki oldaluk, ahol az általuk nyújtott szolgáltatások kiválasztásához, beüzemelésükhöz és használatához rengeteg útmutató található.

A megrendelt tárhelynek csak a mérete van megszabva. Korlátlan adatforgalom, korlátlan FTP felhasználói hozzáférés és korlátlan levelező fiók áll rendelkezésre az év 99.9 százalékában. Ezek a feltételek is az EZIT Kft. szolgáltatásainak választása mellett szóltak.

2.3.1 cPanel

A bérelt webtárhelyem linux alapú webkiszolgáló rendszere a cPanel, melynek a 70.0.47-es verziószámú kiadását használom. Ezen a felületen láthatom a tárhelyem lemezhasználatát, processzorhasználatát, fizikai memória kihasználtságát, input/output leterheltségét és ezek felső korlátját. A főoldalon hozzá tudom rendelni a használni kívánt domain címeket, követhetem az előző bejelentkezéseimet, hozzáadhatok könyvtárakat a tárhelyemhez, ftp fiókokat és ezeket módosíthatom is a megfelelő jogosultságokkal. Állíthatok be biztonsági mentést, itt hozom létre az adatbázisokat és hozzá tartozó felhasználókat, itt állíthatom be az email fiókokat és hozzá felhasználókat, létrehozhatok aldomaineket, tehetek be IP-cím tiltásokat és egyéb biztonsági intézkedéseket, lekérdezhetek látogatói és terheltségi statisztikákat. A cPanel biztosítja a teljes körű testre szabhatóságot és kezelést a web alkalmazásom üzemeltetése körül. Továbbá innen érem el a phpMyAdmin oldalt is, ahol az adatbázison belüli műveleteket kezelem, táblák létrehozása, módosítása, feltöltése.

2.4 Operációs rendszer

Szakdolgozatom elkészítése alatt jogtiszt Windows 10 Professional 1809-es verziójú operációs rendszer állt rendelkezésemre, mely megbízhatóan, hibamentesen működött mindvégig.

3 Fejlesztői dokumentáció

A fejlesztésem megkezdése előtt először a főoldal megjelenésére készítettünk tervet. Eleinte hagyományos elrendezésben gondolkoztunk, de a modern, letisztult és reszponzív [3] megjelenés mellett döntöttünk. Átbeszéltünk valamennyi információt, tartalmat, amit mindenképp a főoldalon szeretnénk megjeleníteni. Ez után a cég működését vettük górcső alá, hogy milyen folyamatai vannak, milyen lépésekből állnak, hogyan és hol vezetik ezeket, van-e olyan lépés, ami elhagyható, tudunk-e egyszerűsíteni folyamatainkon. Innen jött az ötlet, a fő modulja az alkalmazásomnak, az időpontfoglalás webes megoldása. Itt ugyancsak szóba került a sokat emlegetett biztonság kérdése, aminek megoldásán a megbeszélésen kívül hosszasan utánajártam. Az időpontfoglalás egyszerűnek hangzik, de minél többet gondolkodtunk rajta, hogyan lehetne a legjobban automatizálni és emellett mégis minden résztvevőt értesíteni úgy, hogy közben a webalkalmazás látogatójának is megmaradjon a

módosítási joga, illetve biztosnak érezze magát a szolgáltatás minőségével és precizitásával kapcsolatban. Mindezek után tervezésbe kezdtem, hogyan működjön a rendszer, mi és hogyan lássa el a feladatokat, milyen struktúrában tároljam az adatokat, mennyi adatra is van szükségem, miket kell az ügyfelekkel feltöltnem és mi az, ami már a cég munkatársainak a feladata. Sok kérdés merült fel bennem, amikre a választ a következő szekciókban megadok.

3.1 Rendszer komponensei

- index.html
- logon.html
- registration.html
- megerosites.html
- admin_booking.php
- admin_booking_clear.php
- admin_booking_close.php
- admin_booking_del.php
- admin_booking_upload.php
- admin_booking_workers.php
- admin_workers.php
- admin_workers_del.php
- booking.php
- booking_del.php
- booking_details.php
- confirm.php
- contact.php
- logon.php
- logout.php
- registration.php

3.2 Adatbázis terv

A webalkalmazás adatbázisában, ami a Qualityw_online (1. ábra) főként a felhasználók regisztrációs adatait tároljuk legfőbb paramétereikkel, a cégnél dolgozó alkalmazottak adatait, az időpontokhoz tartozó adatokat, valamint mindezek összekötését egy kapcsolótáblával valósítom meg. Az összes táblában odafigyeltem az adattípusok fajtáira és méreteikre is, hogy a továbbiakban megfelelően használható legyen a programfutás során.

3.2.1 Adatbázis táblák

Users: Az alkalmazás összes felhasználójának adatait ebben a táblában tárolom. Minden felhasználónak van egy *id*-ja, ami egy 1-től induló automatikusan növekvő számsorozat, ez a táblában generálódik. A regisztrációs form kitöltésével a látogató feltöltheti a teljes nevét – *name* -, az email címét – *email* -, ami a továbbiakban a regisztráció megerősítésénél, bejelentkezésénél és további rendszerüzenet fogadásánál rendkívül fontos, telefonszámát – *phone* – és lakcímét – *address* -. Továbbiakban még van egy 1 karakteres *enable* flag, ami a regisztráció visszaigazolása után lesz engedélyezve (1), alapértelmezetten *null*. *Admin* flag, ami az oldal felhasználóit (0) és a vállalkozás adminisztrátorait (1) különbözteti meg. Ezt a bit-et is ellenőrzi a legtöbb programrész, ez alapján jelennek meg az adminisztrátori funkciók. *Last_login*, ahol a felhasználó legutolsó bejelentkezését tudjuk nyomon követni. *Reg_date*-ben a regisztráció dátumát tároljuk. Végül a *confirm_hash*, ami az *enable* mezővel szorosan együtt dolgozik. Regisztrációkor generálódik egy a hash karaktersorozat és ha ugyanerre hivatkozik a felhasználó az email visszaigazolós linken keresztül, akkor lesz engedélyezve a felhasználója, majd ez a hash sorozat törlődik a táblából.

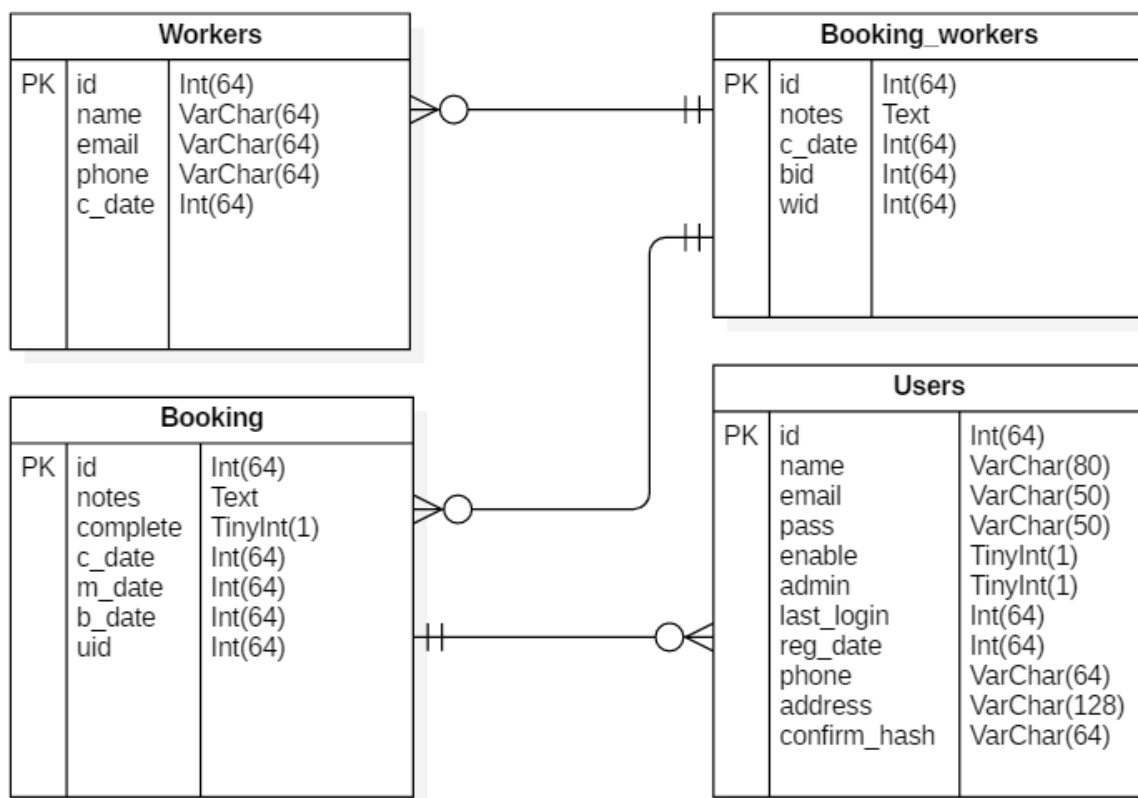
Workers: A vállalat dolgozóinak, szerelőinek az adatait tároljuk ebben a táblában. Az *id* mezőben szerepel a dolgozó azonosítószáma, ami szintén felvételkor generálódik. *Name* mezőben szerepel a teljes neve, *email* mezőben az email címe, a rendszerüzenetek miatt fontos. *Phone*, telefonszám és *c_date*-ben tároljuk a dátumot, amikor felvitték a rendszerbe a dolgozó rekordját.

Booking: Az időpontfoglaló modul egyik táblája. Az *id* itt a foglalható időpont egyedi azonosítója. Minden időszletnek külön egyedi azonosítója van, függetlenül attól, hogy foglalt-e vagy sem. *Notes* mezőben tudunk megadni az időpont lefoglalásakor megjegyzéseket, hogy milyen szolgáltatással kapcsolatban foglaltunk időszletet. *Complete*,

ez egy bit, ami alapértelmezetten *null*, egy megrendelés lezárásakor változik 1-re, jelezvén, hogy teljesült a megrendelő kérése, megtörtént az interakció a dolgozók részéről. *C_date*-ben a szabad időpont rögzítésének a dátuma szerepel, *m_date*-ben az időpont módosításának dátuma és *b_date*-ben pedig az időpont lefoglalásának dátuma. Amikor egy időpontot lefoglalnak nem csak a *b_date* változik, hanem az *uid* mezőbe is bekerül a felhasználó azonosítója, aki lefoglalta. A **Booking**(*uid*) egy kapcsolat a **Users**(*id*) mezővel.

Booking_workers: Ez egy kapcsolótábla. Amikor egy időpontot lefoglal egy felhasználó, akkor hozzáadódik a foglaláshoz az azonosítója, magyarul hozzá kapcsoljuk a felhasználót az időponthoz. Szerettem volna, ha nem csak a felhasználót, hanem a dolgozót is hozzá lehetne kapcsolni egy adott munkához. Ehhez elég lehetett volna még egy mező a **Booking** táblában, de felmerült egy kérdés, hogy jó lenne, ha egy munkához hozzá lehetne rendelni akár több dolgozót is. Emiatt készült el ez a tábla, ami N:1-hez kapcsolat a felsorolt módon: **Booking**(*id*) – **Booking_workers**(*bid*) és **Workers**(*id*) - **Booking_workers**(*wid*). Itt is jelen van egy *id*, azonosító mező, *notes*, megjegyzés mező. *C_date*, dátum mező, ahol a dolgozó hozzárendelésének dátuma van tárolva. Végül a fentebb említett két azonosító mező *bid*, mint **Booking**(*id*) és *wid*, mint **Workers**(*id*).

3.2.2 Relációs adatmodell



1. ábra Relációs adatmodell, a Qualityw_online adatbázis ER diagrammja

3.3 A rendszer felépítése

A következőkben bemutatom a webes alkalmazásom *frontend* és *backend* felépítését, moduljait, valamint a felhasználó szemszögéből magát a felhasználói felületet, hogy milyen oldalakból, lapokból áll az alkalmazás. Ezeken az oldalakon milyen funkciók érhetőek el számukra és a weboldal elemeit ismertetem. Itt kiegészítve az általános felhasználói funkciók mellett az admin funkciókkal, nézetekkel is. Másik oldalról (*backend*) pedig a modulok, funkciók működését ismertetem, milyen technológiákat használtam, hogyan valósítottam meg az ötletektől a gyakorlatig az egyes feladatokat. Ezeken felül kitérek még fontos témákra, amiknek utánajártam, hogy minél korszerűbb terméket adhassak át, ami az esetleges jövőbeli fejlesztések esetén is megállja a helyét és a rendszer biztonságtechnikai szempontjait is kiemelném, amiknek megfelel.

3.3.1 Felhasználói felületek

Tematikusan három részre lehet bontani a felhasználói felületet. A QualityWorld Kft. főoldala az első, amit keresőóriásból, egyéb hirdetésekéből vagy közvetlenül elérhet bárki. Magával a főoldallal és a rajta levő tartalommal, bemutatkozással minden látogató találkozni fog. A második, az alkalmazás egyik fő funkciója, az időpontfoglaló modul. Amivel csak azok az érdeklődők fognak találkozni, akik szeretnének a cégtől online szolgáltatást igénybe venni. Ez egy felületre irányít át, ugyanúgy a QualityWorld *domain* tartományán belül. Itt a naptár nézettel, időpontok foglalásával, kezelésével fogunk találkozni, de ennek biztonságos és megfelelő működése már megkövetel egy regisztrációs/bejelentkezős rendszert is. Végül a harmadik a foglalásokkal kapcsolatos adminisztrációs felület, ami a látogatók számára rejtett marad mindvégig. Ez a cég asszisztense, vezetője számára lett fejlesztve, hogy a felhasználói foglalásokról legyen egy összesített nézete, könnyebben fel tudja venni a kapcsolatot a leendő megrendelőkkel és menedzselés szempontjából további funkciókkal egyszerűsítsék folyamataikat.

3.3.1.1 Oldalak listája

- Főoldal
- Bejelentkező oldal
- Regisztrációs oldal
- Email megerősítő oldal
- Időpontfoglaló felület
- Időpontfoglaló / feltöltő felület (Admin)
- Foglalt időpontok nézet (Admin)
- Szerelő hozzárendelése nézet (Admin)
- Rögzített szerelők / feltöltő felület (Admin)

3.3.1.2 Funkciók részletes leírása

Főoldal: a *Qualityworld.hu domain* betöltése után megjelenik előttünk a cég főoldala [6]. Felül egy navigációs sáv található, (balról jobbra) rajta egy hivatkozással a főoldalra, majd négy menüpont, amik ugyan ezen a weboldalon tartanak, csak a tematikus részek közötti váltásra szolgálnak, mint gyorsgomb. Tehát a *Cégünk*ről gomb leugrik a *Cégünk*ről pont tetejéhez, a *Szolgáltatásaink* odairányít a *Szolgáltatásaink* rész elejére és így tovább a

Munkáink és *Kapcsolat* gomb. A menüsáv kinézete változik az eszköz megjelenítőjének kijelző paramétereitől, felbontásától és az eszköz jellegétől (számítógép, tablet, okostelefon) függően. Így, ha egy kisebb kijelzőjű eszközön nyitjuk meg az oldalt, akkor csak a QualityWorld hivatkozás fog megjelenni és egy menü jelzés (három vízszintes vonal), amire kattintva lenyílik a fentebb ismertetett négy menüpont hivatkozása, újbóli kattintásra pedig visszagördül, eltűnik. A menüsáv mindvégig az oldal tetején megtalálható bármeddig is görgetünk és színváltozással jelzi, hogy épp hol is tartunk a menüpontok között. A sáv alatt látható a hatalmas fejléc, rajta a cég megnevezésével, egy rövid szlogennel és egy ajánlatkérő gombbal, ami átirányít a következő felületre. Erről majd annál a résznél írok bővebben. Lejjebb egy képnézegető, diavetítő található, ami automatikusan lapoz pár másodpercenként a képek között. Indikátorokkal jelzi, hogy hányadik képnél tart, az utolsó kép után előlről kezdi a vetítést, végtelenítve, illetve, ha a kép felett van a kurzor, addig nem léptet a következő slide-ra. Kézzel is lehet léptetni előre-hátra. Elértünk a *Cégünk*ről szekcióig, ahol pár soros bemutatkozás található, majd, ha kedvet kap az oldal látogatója, a sorokat olvasva azonnal oda is ugorhat a kapcsolat felvételi szekcióhoz egy *Kapcsolat* gombbal. Következő szekció a *Szolgáltatásaink*, itt a szolgáltatások egy része megtalálható felsorolva, illetve alatta pár minősítés kis ikonokkal jelölve. Az oldal menüsávja, fejléce, ikonjai és vezérlői mind egy színkóddal vannak ellátva, melyet módosítva könnyen változtatható a cég arculata, az oldal egységes megjelenése. A *Munkáink* résznél pár képpel, esetleg képaláírással is bemutatható egy két referencia. Végül, de nem utoljára a *Kapcsolat* szekció. Itt bal oldalon látható a cég földrajzi helye, egy aktuálisan hívható telefonszám és egy email cím, beépített hivatkozással. Erre kattintva átirányít az oldal az eszközön beállított elsődleges levelező kliensbe és kitölti a címzett mezőt a hivatkozáson is megjelenített email címmel, ami a vállalkozás egyik fő kapcsolat felvételi, értesítő email címe. Jobb oldalt található egy kis kitölthető *form*. Név, email cím és egy üzenet megadása után a küldés gombra kattintva újratöltődik az oldal. Természetesen az újratöltésen kívül ezzel egy üzenetet tudunk küldeni a megfelelő címre, ahonnan mihamarabbi válaszra számíthatunk. Hogy biztosak legyünk az elküldés sikerességében nem csak a cég egy közös email címére érkezik meg az üzenetünk, hanem mi is kapunk egy másolatot az üzenet tartalmával a *form*-on megadott email címre. Végül a lábléc található a weboldal alján, ahol a szerzői jog tulajdonosa olvasható, az aktuális évszámmal, alatta egy hivatkozással ugyancsak a főoldalra és felette egy kék felfele nyíllal. Fölé húzva a kurzort beszédes üzenet jelenik meg, a nyílra kattintva felugrik az oldal tetejére.

Bejelentkezés: Ha az oldal látogatói amellettt döntenek, hogy szimpatikus a cég és megrendelnék valamely szolgáltatásukat, akkor a fejlécben szereplő *Kérjen ajánlatot még ma!* gombra kattintva indulhatnak el a következő, tárgyban szereplő funkcióhoz. Ekkor először a jelentkező felület jelenik meg, ahol email és jelszó mező tűnik szembe felül, lentebb három gomb. Email cím és jelszó *Captcha* megadása után a jelentkezésre kattintva beléphetünk az alkalmazásba, ha már van profilunk, ellenkező esetben hibát kapunk. Ha még nem rendelkezünk aktív felhasználói fiókkal a QualityWorld Kft. rendszerében, akkor a *Regisztráljon!* gombra kattintva készíthetünk egyet. Ezzel a kattintással átirányít minket a regisztrációs oldalra. Itt öt mező vár kitöltésre, *Teljes név, Email cím, Telefonszám, Lakcím és Jelszó*. Mindegyik mező kitöltése kötelező a regisztráció elküldéséhez (*Regisztráció* gomb). Ha csak véletlen navigáltunk a regisztrációs oldalra, akkor van egy *Vissza a jelentkezéshez!* gomb is, ami értelemszerűen visszairányít az előző, jelentkező oldalra. Amint elküldjük a regisztrációkat, akkor megjelenik egy információs oldal, ahol felhívják a figyelmünket az email megerősítésre és probléma esetén szerepel egy email elérhetőség is (itt is *mailto* funkcióval). Email fiókunkat megtekintve ott vár minket a regisztrációs rendszerüzenet, a benne található információkkal, instrukciókkal és a megerősítő hivatkozással, linkkel, amire kattintva megtörténik a regisztrált fiókunk aktiválása, majd újra a jelentkező felületen találjuk magunkat. Még mielőtt megpróbálnánk jelentkezni, van egy utolsó gomb is, *Vissza a főoldalra!*, ezzel visszaléphetünk a főoldalra, ha csak véletlen vagy kíváncsiságból léptünk volna az időpontfoglalásra, esetleg még gondolkodnánk. Most az aktivált email cím jelszó párossal jelentkezhetünk és elénk tárul az időpontfoglaló felület.

Időpontfoglalás: Az időpontfoglalás nézet egy oldalból áll a felhasználók részére. A főoldal tetején köszöntjük a felhasználót a teljes nevével, mellette hivatkozásként megjelenik a *Kijelentkezés* gomb, amivel kiléphetünk a fiókunkkal ebből a rendszerből. Alatta megjelenik a várva várt *Időpontfoglalás* felülete, majd dátum formátumban mindig az aktuális hét hétfője és kötőjellel elválasztva az aktuális hét vasárnapja. Ez a két dátum külön-külön egy-egy hivatkozás, funkciókkal ellátva. Az előbbi dátumra kattintva az előző hét hétfői és vasárnapi dátuma fog megjelenni a helyükön, utóbbira kattintva pedig a következő hét hétfő és vasárnap párosa. Ezzel párhuzamban vezéreljük az alatta található naptárat is. A naptár táblázat szerű elrendezésben jelenik meg, ahol a fejlécében szerepel az adott nap megnevezése hétfőtől vasárnapig, alatta pedig az adott nap dátuma és ha lapozzuk a heteket előre hátra, akkor az itt szereplő dátumok is ennek megfelelően változnak, illetve a lent

található időpontok is. Mindegyik időpont (tól-ig) háttérszínében elkülönül a többitől, ha elhúzzuk felette a kurzort és mindegyik önmagában egy külön hivatkozás. Bármelyik időpontra kattintva egy új felületre irányít bennünket, ahol jól láthatóan megjelenik a foglalni kívánt dátum kezdő időpontja és időtartama, alatta a felhasználó adatai és egy megjegyzés mező. Ezt a mezőt célszerű kitölteni a kívánt szolgáltatás megnevezésével, esetleges adatváltozással és egyéb kéréssel. Ez alapján fogja a cég adminisztrátora felkeresni az érdeklődőt az időpontra foglalt kérés pontosításával kapcsolatban. Az *Időpontfoglalás* gombra kattintva el is küldi a kérésünket a rendszer felé, erről értesítést kap a cég és mi is email formájában, amiben szintén szerepel a pontos időpont. Újra az időpontfoglalás oldalra irányít minket, ahol áthúzással foglalt státuszba is került az időpontunk, ezután más sem tudja újra lefoglalni. A naptár alatt van még a *Foglalt időpontjaim* oszlop. Itt a felhasználó által lefoglalt időpontok adatai találhatóak meg, alattuk a megjegyzés mezővel, amit a foglaláskor megadtunk és minden foglalt időpontunk mellett megjelenik egy törlés gomb. Ezzel fel tudjuk szabadítani a foglalt időpontunkat, törölhetjük foglalásunkat. Ezzel megszűnik a naptárban az időpont áthúzása és újra hivatkozásként működve bárki lefoglalhatja magának az időszakot.

3.3.1.3 Időpontfoglalás adminisztrátor felülete

Ha a cég egy meghatalmazott munkatársa jelentkezik be a rendszerünkbe, akinek a felhasználói fiókján engedélyezve van az adminisztrátori jogosultság, annak a fenti sávban megjelenik egy [Admin] minősítés a neve mellett és további menüpontok, ami egy részletesebb, több funkcióval ellátott felületre utal. Ezeket a funkciókat fogom most ismertetni, amik a következőkből állnak.

Admin – Időpontfoglalás

Első ránézésre megegyezik az egyszerű felhasználói nézettel, de ez nem teljesen igaz. Az aktuális hét naptárnézetének fejlécén minden naphoz került egy [Feltöltés] gomb. Az egyik ilyen gombra rákattintva megjelenik az időpontok feltöltése felület, ahol az adott nap dátuma kiemelten látszik legfelül, alatta pedig megadhatjuk, hogy az adott napon mely időpontok legyenek foglalhatóak. Itt meg kell adnunk egy kezdő időpontot órában, amikortól induljanak a szabad időpontok, a következő mezőben pedig a foglalható időpontok darabszámát. Ha elküldjük a kérést a *Feltöltés* gombbal, akkor az adott napra megjelenik a kezdő időponttól annyi egy órás időpont, ahány darabot szerettünk volna. A program

publikálása és gyakorlatban való használata után lehetőség nyílik még az időpontok finomítására. Jelenleg egy órás szeletekben egyeztünk meg, szinte a legtöbb munkánál ez a minimum időtartam, ami mindenképpen szükséges. Az egy órás időtartam feletti munkáknál pedig az adminisztrátor és a megrendelő között lezajlott egyeztetés, kapcsolatfelvétel alapján lehet a foglaláson módosítani, kibővíteni vagy esetleg áttenni másik napra, ha közvetlenül az időpont előtt vagy után nincsen szabad hely.

Admin – Foglalt időpontok. Itt az aktuális heti időpontok jelennek meg egy oszlopos nézetben, ahol felsorakoznak a szabad időpontok is. Itt is, mint az alap *időpontfoglalás* nézetben tudjuk lapozni a heteket előre-hátra. A szabad időpontokat tudjuk törölni a mellettük szereplő *törlés* hivatkozással, ha pedig egy időpontot lefoglaltak, közvetlen a foglalás mellett megjelenik az ügyfél pár személyes adata (neve, email elérhetősége, telefonszáma) és három szerkesztési gomb (*szerelők*, *felszabadítás*, *lezárás*). Továbbá alatta látható a megjegyzés, amit foglaláskor megadott az érdeklődő, illetve a foglaláshoz felvett szerelők. A funkciókkal folytatva, a *felszabadítás* gombbal egy foglalt időpontról törölni tudjuk a foglalást, ezzel felszabadítva az időpontot. Ügyféllel való egyeztetés után, ha kiderülne, hogy téves foglalás, esetleg módosítani szeretne, szeretnénk a foglalás idején, akkor ezzel a funkcióval tudja az adminisztrátor levenni az időpontról a foglalást. Következő gomb a *lezárás*, amit akkor használunk, ha egy megrendelés vagy az adott időpontra foglalt szolgáltatás sikeresen teljesül, ezzel lezárva a nyitott kérést. A jövőben az így összegyűlt tranzakciókból további hasznos adatokat lehet kinyerni. A *szerelők* gomb átirányít minket egy újabb felületre, ahol a rögzített szerelők közül egyet vagy többet hozzá tudunk rendelni az adott foglaláshoz. Ezt a beállítást a *Szerelők rögzítése* gombbal tudjuk nyugtázni. Ekkor a foglalt időpont mellett meg is jelenik a szerelő vagy szerelők, akiket hozzárendeltünk a munkához. Újra a *szerelők* gombra kattintva megjelenik ez az opció, bepipálva a jelenlegi hozzárendelésekkel, ahol lehetőségünk nyílik módosítani a hozzárendeléseken, bővíteni vagy szűkíteni a hozzárendelt szerelők közül.

Admin – Szerelők kezelése

Ezen a felületen először a rögzített szerelők sorakoznak fel és az ő elérhetőségeik (*név, email cím, telefonszám*). Minden egyes rekord végén szerepel egy *Törlés* gomb, amire rákattintva az adott szerelőt törölhetjük az adatbázisunkból. Ennek következménye, hogy nem csak a listából tűnik el, hanem a foglalásokhoz sem tudjuk hozzárendelni őt többé. Természetesen

valahogy fel is kell vinnünk a szerelőket, itt jelenik meg az *új szerelő rögzítése* szekció, ahol a szerelő nevét, email címét és telefonszámát megadva a *Rögzítés* gombbal felvehetjük az adatbázisunkba. Rögzítés után egyből meg is jelenik a fenti táblázatunkban és egyből hozzárendelhetővé is válik a foglalt időpontokhoz. Az admin felületek között a fenti sávban található gombokkal biztosított a szabad navigálás.

3.3.2 Modulok működése

Ebben a pontban ismertetem a webalkalmazás működéséhez elengedhetetlen kódrészleteket, amikkel megvalósítottam az egyes modulokat. Emellett bemutatom a főoldal felépítését és a benne használt osztályokat is, majd az időpontfoglalás felületét, a naptár és hozzá kapcsolódó funkciók működését, logikáját. A főoldal HTML5 [7] nyelven íródott CSS3 felhasználásával. A bejelentkező, regisztrációs, az időpontfoglalós és ennek admin felülete még ki lett egészítve PHP5 programnyelven készült kódrészletekkel. Az oldal által generált adatokat MySQL5 adatbázisban tárolom. Az oldal reszponzív megjelenítéséhez Bootstrap [3] keretrendszert használtam.

Főoldal – index.html

A Qualityworld.hu domaint betöltve az index.html fájlunkat tölti le először a böngésző, mely további fájlokat hív meg, tölt le. A <head> tag-eken belül adom meg a weboldal címét <title> és hívok meg scripteket, fájlokat, amik a tartalom megjelenéséhez elengedhetetlenek. Ezek például a Bootstrap [3], css és javascript fájljai, az oldalakhoz készített stílus css fájl (style.css) és a weboldal címe mellett megjelenő ikon meghívása. A <body> tag-eken belül használtam több bootstrap osztályt, elsőként a navbar-t, ami a fent mindig elérhető navigációs sávot valósítja meg.

```

20 <body id="myPage" data-spy="scroll" data-target=".navbar" data-offset="60">
21 <nav class="navbar navbar-default navbar-fixed-top">
22   <div class="container">
23     <div class="navbar-header">
24       <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">
25         <span class="icon-bar"></span>
26         <span class="icon-bar"></span>
27         <span class="icon-bar"></span>
28       </button>
29       <a class="navbar-brand" href="#myPage">Quality World</a>
30     </div>
31     <div class="collapse navbar-collapse" id="myNavbar">
32       <ul class="nav navbar-nav navbar-right">
33         <li><a href="#about">Cégünkéről</a></li>
34         <li><a href="#services">Szolgáltatásaink</a></li>
35         <li><a href="#portfolio">Munkáink</a></li>
36         <li><a href="#contact">Kapcsolat</a></li>
37       </ul>
38     </div>
39   </div>
40 </nav>

```

2. ábra Navigációs sáv működése

A navbar-fixed-top osztály használata mellett jelenik meg mindig a navigációs sáv (2. ábra) a weboldal tetején, bárhova is görgetünk rajta. Alatta pedig a sáv tartalmának logikája látható. Bal oldalon mindig a Quality World hivatkozás jelenik meg (navbar-brand osztály), ami a főoldalra navigál, tőle jobbra pedig a menüpontok, amik az oldal egyes részeire navigálnak. Például a kapcsolat gomb a #contact-ra navigál minket, ez a kapcsolat részt körülvevő <div> tag-eknek az ID-ja. Ha pedig a megjelenítőn nem fér ki a navigációs sáv minden eleme (például okostelefon), akkor csoportosítja őket (collapse) egy lenyitható menübe, ami jelen esetben három vonal, tag-ek icon-bar osztályban.

```

53 <!-- SlideShow -->
54 <div id="myCarousel" class="carousel slide text-center" data-ride="carousel">
55   <ol class="carousel-indicators">
56     <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
57     <li data-target="#myCarousel" data-slide-to="1"></li>
58   </ol>
59   <div class="carousel-inner">
60     <div class="item">
61       
62       <div class="carousel-caption">
63         <h3>Samsung</h3>
64         <p>Üzleti megoldások</p>
65       </div>
66     </div>
67     <div class="item">
68       
69       <div class="carousel-caption">
70         <h3>Samsung</h3>
71         <p>Üzleti megoldások</p>
72       </div>
73     </div>
74   </div>
75   <a class="left carousel-control" href="#myCarousel" role="button" data-slide="prev">
76     <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
77     <span class="sr-only">Elozo</span>
78   </a>
79   <a class="right carousel-control" href="#myCarousel" role="button" data-slide="next">
80     <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
81     <span class="sr-only">Kovetkezo</span>
82   </a>
83 </div>

```

3. ábra Slideshow működése

Következő bootstrap osztály a carousel, ami a slideshow működéséért felel. A carousel elemei (<div>-ek) az item osztályt kapják. Mindegy egyes item osztályú <div> egy képet jelen, amire akár szöveget is írhatunk (carousel-caption). A slideshow (3. ábra) állását kör alakú gondolatjelek jelzik (carousel-indicators), a style.css-ben van megírva a színváltozása, mindig az éppen aktív (active) osztályú gondolatjel kap háttérszínt, ezáltal válik teli körre. Még jobb és bal oldalt van egy-egy nyíl (left/right carousel-control), amikkel kézzel is lapozhatjuk a képeket. Természetesen a slideshow folyamatos, tehát az utolsó kép után az elsőre ugrik, ugyan ez vonatkozik az indikátorokra is.

A fő oka amiért beépítésre került a **Bootstrap** framework (keretrendszer) az a webalkalmazás reszponzív megjelenésének biztosítása. Használatához a <head> tag-ek között hozzá kell adnunk egy CSS fájlját, több javascript könyvtárát és a reszponzív viewport meta tag-et.

Ahogy a navigációs sávnál bemutatam, keskenyebb megjelenítőknél csoportosítja a sáv tartalmát egy lenyitható menübe és ott nem vízszintesen, hanem függőlegesen nyílik le logikusan. A Bootstrap egy rácsszerkezet (grid system) használatát teszi lehetővé, ahol egy sorban (row osztály) 12 elem (col) férhet el egymás mellett, de ezeknek az elemeknek a méretét, egymáshoz képesti arányát, az oldalon való elosztását osztályokkal tudjuk szabályozni. Így ami egy asztali számítógép megjelenítőjén kifer egy sorba, az egy mobil eszközön (keskenyebb megjelenítőn) egymás alá rendezve fog megjelenni attól függően, hogy hány eleme van egy sornak, de az elemek egymáshoz képesti méretarányait tartva. Azonban, ha egy sorban csak egy elem szerepel, akkor azt lekicsinyíti addig, hogy a keskenyebb megjelenítőn kiferjen, mivel nincs felosztva így nem tudja hogyan tördelje egymás alá. A rácsszerkezet elemeinek a maximum szélességét hat méretben tudjuk megadni, amit ugyanúgy az osztályban jelölünk a következőképp:

- col – automatikus
- col-xs – ha a kijelző szélessége kisebb mint 768 pixel, okostelefonokra vonatkozhat, ekkor maximum 540 pixel széles lehet az elem mérete
- col-sm – ha a kijelző szélessége nagyobb vagy egyenlő mint 768 pixel, táblagépekre vonatkozhat, ekkor maximum 720 pixel széles lehet az elem mérete
- col-md - ha a kijelző szélessége nagyobb vagy egyenlő mint 992 pixel, kisebb laptopokra vonatkozhat, ekkor maximum 960 pixel széles lehet az elem mérete

- col-lg - ha a kijelző szélessége nagyobb vagy egyenlő mint 1200 pixel, általános laptop vagy asztali számítógépek esetén, ekkor maximum 1140 pixel széles lehet az elem mérete
- col-xl - ha a kijelző szélessége nagyobb vagy egyenlő mint 1440 pixel, nagyobb megjelenítőknél alkalmazandó

Ezeknek az osztályoknak az összekombinálásával még dinamikusabb oldalt tudunk létrehozni. Továbbá a sorokon belüli elemek méretarányát is az osztályban tudjuk megadni. Erre példaként a főoldal alján található kapcsolat részt (4. ábra) szeretném bemutatni.

```

200 <!-- Kapcsolat -->
201 <form action="/.php/contact.php" method="POST">
202   <div id="contact" class="container-fluid bg-grey">
203     <h2 class="text-center">Kapcsolat</h2>
204     <div class="row">
205       <div class="col-sm-5 left">
206         <p>Kérdéseire szívesen válaszolunk. <br> Keressen minket bátran!</p>
207         <p><span class="glyphicon glyphicon-map-marker"> 2400, Dunaújváros</p>
208         <p><span class="glyphicon glyphicon-phone"> +3620/359-7519 </p>
209         <p><span class="glyphicon glyphicon-envelope"></span><a href="mailto:info@qualityworld.hu">
210           info@qualityworld.hu</a></p>
211       </div>
212       <div class="col-sm-7">
213         <div class="row">
214           <div class="col-sm-6 form-group">
215             <input class="form-control" id="name" name="name" placeholder="Név" type="text" required>
216           </div>
217           <div class="col-sm-6 form-group">
218             <input class="form-control" id="email" name="email" placeholder="E-mail cím" type="email" required>
219           </div>
220         </div>
221         <textarea class="form-control" id="comments" name="comments" placeholder="Miben segíthetünk?" rows="5"></textarea><br>
222         <div class="row">
223           <div class="col-sm-12 form-group">
224             <button class="btn btn-default pull-right" type="submit">Küldés</button>
225           </div>
226         </div>
227       </div>
228     </div>
229   </form>
230 </div>
231 </div>
232 </form>

```

4. ábra Kapcsolatfelvételi modul felépítése

Ez egy teljes form, ami submit action-re (küldés gomb megnyomására) meghívja a ./php/contact.php fájlt és POST metódussal értékeket ad át a PHP program részére, jelen esetben name, email, comments input mezőbe bevitt értékeket. Beljebb <div> tag ID-ja a contact, erre hivatkozunk a navigációs sávban, innen tudja, hogy a képernyő tetejéhez melyik <div> tag kezdetét kell igazítani. A divíziót (<div>) egy sorként kezeljük, amit kettő mezőre (column) osztunk 5:7 arányban (szélességük arányainak összege 12), ami úgy tevődik össze, hogy a bal oldali részt col-sm-5 osztály jellemzi, míg a jobb oldalt col-sm-7 osztály. Bal oldalt kapcsolatfelvételi adatok jelennek meg és a jobb oldalon pedig a kapcsolatfelvételi form. A kapcsolatfelvételi formot kettő külön sorként kezelem. Az egyik sorában 6:6

arányban oszlik fel a kettő input mező és alatta pedig egy teljes szélességű öt soros textarea. A következő sorba áttettem a submit gombot, hogy ne rontsa el a többi mező összhangját. Tehát a kijelző teljes szélességét 5:7 arányban felosztottam, majd a jobb oldali teljes grid szélességből maradt hetet is tovább feleztem a kettő input mezőnek és mivel feleztem ezért új sorba kényszerül tördelni mindig a nagyobb textarea mezőt, emiatt jelenik meg mindig kényszerítve alattuk. A gomb pedig az új row osztály miatt került egy sorral lejjebb. Így működik a Bootstrap grid rendszere. Sokféleképpen lehet kényszeríteni, hogy a megjelenést tartsa, de mégis optimalizált legyen megjelenítő méretétől függetlenül.

Bejelentkezés – logon.html, logon.php

A főoldal fejlécében található ajánlatkérő gomb átirányít minket egy logon.html fájlra, amin egy form található. Email cím és jelszó kitöltése után a bejelentkezés gombra kattintva (submit) meghívódik a ./php/logon.php és POST metódussal átadjuk az email és password mezőbe írtakat. A logon.php (5. ábra).

```
9  if(isset($_POST['email']) && isset($_POST['password']) && !empty($_POST['email']) && !empty($_POST['password']))
10  { //Ha a login formon az email és a password inputok léteznek és nem üresek, akkor
11
12      //MySQL kapcsolódás
13      $host = "localhost:3306";
14      $dbuser = "qualityw_system";
15      $password = " ";
16      $dbname = "qualityw_online";
17
18      $conn = mysqli_connect($host,$dbuser,$password,$dbname);
19
20      //Lekérdezzük és beazonosítjuk a keresett felhasználót
21      $q = mysqli_query($conn,"SELECT * FROM users
22      WHERE email = '".mysqli_real_escape_string($conn,$_POST['email'])."'
23      AND pass = '".mysqli_real_escape_string($conn,md5($_POST['password']))."'
24      AND enable = 1
25      ");
26
27      $user = array();
28      while($q_r=mysqli_fetch_assoc($q))
29      {
30          $user = $q_r;
31          setcookie("uid", $user['id'], time()+3600); /* lejáratí idő 1 óra */
32
33          //Átirányítjuk a felhasználót a foglalási oldalra
34          header('Location: http://qualityworld.hu/php/booking.php');
35      }
36  }
37
38  }else{
39      header('Location: http://qualityworld.hu/logon.html');
40  }
```

5. ábra bejelentkezés osztály

Először megvizsgáljuk, hogy létezik-e, megkaptuk-e POST metódussal az email és password mezők értékeit, ha ezek megvannak és nem üresek, akkor folytatódik a programunk. Esetleg, ha nem teljesülne ez az összetett feltétel, annak a különben (else) ágába csak egy visszairányítás van a logon.html-re, de ez csak akkor fordulhat elő, ha nem kapja meg a logon.php a bejelentkezési adatokat (email, password), mivel a html-ben is ellenőrzöm, hogy ki vannak-e töltve az input mezők. Ott mind a kettő input mező kötelező (required) tulajdonsággal van ellátva, tehát, ha üresen szeretnénk submit-álni, akkor egy figyelmeztető buborék ugrik fel, illetve, ha nem megfelelő formátummal töltönnék ki, akkor is figyelmeztetést dob, mert form-control osztály és input type paraméter párosából tudja, hogy az első mezőnél mik az email követelmények, mi az emailcím formátuma. Tehát, ha érvényesül az első feltételünk, akkor tovább folytatódik a programunk. Azért tettem ezektől a feltételektől függővé a program futását, mert a következő lépésben már kapcsolódik az SQL adatbázishoz, majd lekérdez egy teljes táblát és keresést futtat rajta, amivel feleslegesen ne terhelje a szervert, ha például üres az egyik változóm, abban az esetben úgysem találná meg a felhasználót, ráadásul hibás eredményt is kapnánk. Ahogy írtam, megtörténik a mysql kapcsolódás a *mysqli_connect()* függvénnyel. A szerver és bejelentkezés adatait külön változókba tettem és azokkal paramétereztem a függvényt. A *\$host* a MySQL [1] szerver elérési útvonala, mivel ugyan azon a webszerveren fut, így *localhost*-on elérhető. *\$dbname* az adatbázis neve, amihez csatlakozni szeretnénk és ennek az adatbázisnak az egyik felhasználójával, aminek nevét a *\$dbuser* változóban és annak jelszavát pedig a *\$password* változóban tárolom. A *\$conn* változóban (connect) tárolom az SQL kapcsolatot a *mysqli_connect()* függvényt felparaméterezve a csatlakozási adatokkal, így egy változót elég használnom a lekérdezéseknél a kapcsolódáshoz. *\$q* változóban (query) tárolom az adatbáziskapcsolódást a teljes lekérdezéssel. Ebben az esetben le kell kérdeznünk, hogy a megadott bejelentkezési adatok helyesek-e. Ez a lekérdezés úgy tevődik össze, hogy a *qualityw_online* adatbázisának *users* táblájában keresem azokat a sorokat, ahol az *email* mező egyezik a POST által átadott *email* értékével, a *pass* mező egyezik a POST által átadott *password* értékével és az *enable* mező értéke egyenlő 1-gyel, az utolsó flag értéke akkor 1, ha megerősítette a regisztrációját a felhasználó. Létrehozunk egy *\$user* tömböt majd a *\$q_r* asszociatív tömbbe fetch-eljük az eredményt *mysql_fetch_assoc(\$q)* függvénnyel, aminek paramétere az SQL lekérdezésünk eredménye.

Ezt átesszük a `$user` változóba és állítunk be egy cookie-t a következő függvénnyel és paramétereivel, `setcookie('uid',$user['id'], time()+3600)`, ami létrehozza a sütit, aminek `uid` értéke egyenlő lesz a `users` táblából kinyert rekord asszociatív tömbből a `$user` tömbbe mentett `'id'` értékével és 3600 másodpercig lesz érvényes, 1 óráig.

```
9  if(isset($_COOKIE['uid']) && !empty($_COOKIE['uid']))
10  { //Ha a süti létezik amit a login script mentett, akkor be vagyunk lépve
11
12      unset($_COOKIE['uid']);
13      setcookie('uid', null, -1, '');
14  }
15  header('Location: http://qualityworld.hu/logon.html');
16  ?>
```

6. ábra kijelentkezés funkció

Ehhez kapcsolódóan a kijelentkezés menüpont (6. ábra) megvizsgálja, hogy létezik-e süti. Ha létezik, akkor kiüríti.

Regisztráció – registration.html, registration.php

Ha még nincs felhasználói fiókunk, akkor a bejelentkezés képernyőn a regisztrációra kattintva betöltődik a registration.html oldal. Ez egy kötelezően kitöltendő regisztrációs form, ahol meg kell adnunk a teljes nevünket, emailcímünket, telefonszámunkat, lakcímünket, egy általunk választott jelszót és be kell pipálnunk a nem vagyok robot checkbox-ot, ha sikeresen el szeretnénk küldeni a regisztrációnkat.

```
<script src='https://www.google.com/recaptcha/api.js'></script>
<div class="g-recaptcha" data-sitekey="6Lfwj... nG-zKI3y7"></div>
```

7. ábra Google reCAPTCHA HTML beépítése

Ez az ellenőrzés a google által fejlesztett reCAPTCHA v2 modul (7. ábra). Egy google fiók regisztrációja után egyedi kóddal lehet generálni ezt a beépülő modult. A `<head>` tag-ek közé kellett hozzáadnunk a hozzá tartozó javascriptet és én a form-on belül helyeztem el a `g-recaptcha` osztályú divíziót. Ha a mezők ki vannak töltve és rákattintunk a regisztráció gombra, akkor meghívódik a registration.php fájl és POST metódussal átadjuk az input beviteli mezőkbe írt értékeket.

```

9  $response = $_POST["g-recaptcha-response"];
10 $url = 'https://www.google.com/recaptcha/api/siteverify';
11 $data = array(
12     'secret' => '6LfWyr          :WZKpP',
13     'response' => $_POST["g-recaptcha-response"]
14 );
15 $options = array(
16     'http' => array (
17         'method' => 'POST',
18         'content' => http_build_query($data)
19     )
20 );
21 $context = stream_context_create($options);
22 $verify = file_get_contents($url, false, $context);
23 $captcha_success=json_decode($verify);
24 if ($captcha_success->success==false) {
25     echo "Pipáld be a 'Nem vagyok robot' mezőt!
26     <br><a href='http://qualityworld.hu/registration.html'>Vissza a regisztrációhoz!</a>";
27 } else if ($captcha_success->success==true) {
28

```

8. ábra Google reCAPTCHA szerver oldali php működése

Amit a HTML-ben elhelyezett recaptcha modul átadott POST metódussal a PHP [2] szerver oldali programnak (8. ábra), azt a response kódot összehasonlítja a google által adott titkos (secret) kódunkkal és ha a verifikáció eredménye negatív, abban az esetben hibaüzenetet kap a látogató, azt feltételezve, hogy nem pipálta be ezt a nem vagyok robot checkbox-ot. Ha bepipálta és nem próbálta egyéb módon megkerülni ezt a modult, akkor a PHP elején levő verifikáción át fog jutni, ezáltal a PHP programunk else ágára jutunk és folytatódik a program futása. Megvizsgáljuk, hogy a form mezői ki voltak-e töltve és ha igen, akkor csatlakozunk az adatbázishoz, ugyan úgy, mint a bejelentkezésnél.

```

42 //Lekérdezzük hogy létezik-e a keresett felhasználó
43 $q = mysqli_query($conn,"SELECT * FROM users WHERE email = '".mysqli_real_escape_string($conn,$_POST['email'])."'");
44
45 $van=false;
46 while($q_r=mysqli_fetch_assoc($q))
47 {//ha van, akkor nem adjuk hozzá újra
48     $van=true;
49 }
50
51 if(!$van)
52 {//Ha nincs mentett felhasználó ezen az email címen, akkor rögzítjük
53     $hash = md5(rand(20,500));
54     $id = mysqli_query($conn,"INSERT INTO users (email,pass,name,confirm_hash,reg_date,phone,address)
55     VALUES ('".mysqli_real_escape_string($conn,$_POST['email'])."'
56     , '".mysqli_real_escape_string($conn,md5($_POST['password']))."'
57     , '".mysqli_real_escape_string($conn,$_POST['name']))."', '". $hash."'
58     , '".time()."', '".mysqli_real_escape_string($conn,$_POST['phone']))."'
59     , '".mysqli_real_escape_string($conn,$_POST['address']))."'");
60
61 /*Mail*/
62 $felado="info@qualityworld.hu <info@qualityworld.hu>";
63 $fejlec= "MIME-Version: 1.0\r\n";
64 $fejlec.="From: ".$felado."\r\n";
65 $fejlec.= "Content-Type: text/html; charset=UTF-8\r\n";
66 mail($_POST['email'], "Megerősítő email", "Tisztelt Ügyfelünk!<br><br>Köszönjük regisztrációját! Kérjük erősítse meg regisztrációját
67
68     header('Location: http://qualityworld.hu/megerosites.html');
69
70 }else{echo "Ez az email cím már szerepel az adatbázisban. Kérem adjon meg másik email címet. ("'.mysqli_real_escape_string($conn,$_POS
71 }
72 }
73 ?>

```

9. ábra regisztráció logikája

Itt lekérdezzük, hogy létezik-e már az adatbázisunkban a megadott emailcím, mivel ez is egy egyedi azonosítója a felhasználónak. Ebben a lekérdezésben (9. ábra) is már használom a *mysqli_real_escape_string()* függvényt biztonságtechnikai szempontból. Mivel az SQL lekérdezésembe beágyazok egy felhasználó által megadott string-et, ami akár lehetne SQL kód is, paramétereken keresztül befolyásolni tudnák a lekérdezésem és annak megjelenését. Ez egy kisebb védelmet ad az SQL injection típusú támadások ellen. Tehát, ha szerepel a megadott emailcím már az adatbázisunkban, akkor egy kisebb üzenet jelenik meg a képernyőn, hogy már ezt az emailcímet már valaki felhasználta, kérjük válasszon másik címet és kiíratja a megadott címet, majd egy hivatkozást, amivel visszaléphetünk a regisztrációs form-hoz. Ha még nem szerepel az adatbázisban, akkor generálunk egy hash kódot random számokból, *md5* titkosítással átalakítva és rögzítjük egy INSERT SQL utasítással a megadott adatokat. A users táblába feltöltjük a rekord megfelelő mezőit (email, pass, name, confirm_hash, reg_date, phone, address). A generált hash kód, amit feltöltöttünk az adatbázisba, majd az account aktiválásához lesz szükséges. A reg_date mezőbe az aktuális időt töltjük fel (*time()* függvény). A megadott jelszót titkosítva tároljuk az *md5()* függvény segítségével, a többi mezőbe pedig a felhasználók által kitöltött form értékeit írjuk be megtisztítva a *mysqli_real_escape_string()* függvénnyel. A sikeres INSERT SQL utasítás után a *mail()* függvénnyel küldünk egy emailt az info@qualityworld.hu emailcímről a regisztrált felhasználónak, melyben megköszönjük regisztrációját és egy megerősítő

hivatkozást illesztünk a végére, ami a *confirm.php* fájlra (10. ábra) mutat és a generált *hash* kódot is átadja a hivatkozással. Utolsó lépésként átirányít a megerosites.html oldalra, ahol egy rövid üzenet található további instrukciókkal a regisztrációnk megerősítésével kapcsolatban, probléma esetén egy kapcsolatfelvételi emailcímmel, *mailto* hivatkozásba ágyazva.

Megerősítés – confirm.php

```
19 //Lekérdezzük hogy létezik-e a keresett felhasználó
20 $q = mysqli_query($conn,"SELECT * FROM users WHERE confirm_hash = '".mysqli_real_escape_string($conn,$_GET['hash'])."'");
21
22 $van=false;
23 while($q_r=mysqli_fetch_assoc($q))
24 {
25     $van=true;
26 }
27
28 if($van)
29 {
30     //Ha létezik a confirm hash, akkor aktiváljuk a felhasználót
31     mysqli_query($conn,"UPDATE users SET enable = 1, confirm_hash = ''
32     WHERE confirm_hash = '".mysqli_real_escape_string($conn,$_GET['hash'])."'");
```

10. ábra regisztráció megerősítésének működése

A regisztráció után kiküldött emailben szereplő megerősítő link hivatkozik a *confirm.php* fájlra átadva az egyedileg generált hash kódot GET metódussal. A PHP [2] program (10. ábra) vizsgálja, hogy megérkezett-e a hash kód, ha igen, akkor csatlakozik az adatbázishoz és lekérdezzük a users táblából, hogy létezik-e olyan felhasználó, akinek a *confirm_hash* értéke egyezik a linken kapott kóddal. Ha létezik ilyen felhasználó, akkor konfirmáljuk, aktiváljuk a felhasználói fiókját a következőképpen. Egy UPDATE SQL parancsot futtatunk a users táblán, ami az *enable* flag-et átírja '1'-re és a *confirm_hash* értékét törli, ezzel jelezve, hogy már nem vár aktiválásra az adott felhasználó. Aktiválás után email üzenetet küld a rendszer, hogy sikeresen megtörtént a felhasználói fiókunk aktiválása, a webalkalmazás pedig átirányít a bejelentkező képernyőre, ahol már bátran beléphetünk a rendszerbe, mivel aktív hozzáféréssel rendelkezünk.

Időpontfoglalás – booking.php

Bejelentkezés után az időpontfoglalás oldalára jutunk, ami a *booking.php* fájl. Ettől a résztől PHP nyelven fejlesztettem az oldalakat, a bennük található HTML elemeket echo'' utasítással jelenítettem meg. Az oldal első ellenőrzése, hogy létezik-e a sütiben (COOKIE) tárolt uid (user id), amit bejelentkezéskor hoz létre a rendszer egy órára, ha nem létezik, akkor vissza navigál a bejelentkezés oldalra.

```

37 //Lekérdezzük és beazonosítjuk a keresett felhasználót
38 $q = mysqli_query($conn,"SELECT * FROM users WHERE id = '". $_COOKIE['uid']."'");
39
40 $user = array();
41 while($q_r=mysqli_fetch_assoc($q))
42 {
43     $user = $q_r;
44 }

```

11. ábra a felhasználó adatai sütitől

A sütiten tárolt uid-t beazonosítjuk (11. ábra) az adatbázisunk users táblájában tárolt id-val és eltároljuk a \$user() tömbben a további adatait a felhasználónak.

```

46 //Dátum lekérdezése a címsorból vagy előállítás ha üres
47 if(isset($_GET['datum']) && !empty($_GET['datum'])){$datum=$_GET['datum'];}else{$datum = time();}
48
49 //Hétfőválasztó
50 $het = date("W",$datum);
51 $snap = date("N",$datum); //hétfőtől - vasárnap - 1:7
52 $sev = date("Y",$datum);
53 $honap = date("m",$datum);
54
55 $kalendar = array();
56
57 /* Hétfőválasztó algoritmus */
58 if($snap == 1){$kalendar['kezdes'] = strtotime(date("Y-m-d",$datum)); $kalendar['vege'] = strtotime(date("Y-m-d",$datum)."+6 day");}
59 elseif($snap == 2){$kalendar['kezdes'] = strtotime(date("Y-m-d",$datum)."-1 day"); $kalendar['vege'] = strtotime(date("Y-m-d",$datum)."+5 day");}
60 elseif($snap == 3){$kalendar['kezdes'] = strtotime(date("Y-m-d",$datum)."-2 day"); $kalendar['vege'] = strtotime(date("Y-m-d",$datum)."+4 day");}
61 elseif($snap == 4){$kalendar['kezdes'] = strtotime(date("Y-m-d",$datum)."-3 day"); $kalendar['vege'] = strtotime(date("Y-m-d",$datum)."+3 day");}
62 elseif($snap == 5){$kalendar['kezdes'] = strtotime(date("Y-m-d",$datum)."-4 day"); $kalendar['vege'] = strtotime(date("Y-m-d",$datum)."+2 day");}
63 elseif($snap == 6){$kalendar['kezdes'] = strtotime(date("Y-m-d",$datum)."-5 day"); $kalendar['vege'] = strtotime(date("Y-m-d",$datum)."+1 day");}
64 elseif($snap == 7){$kalendar['kezdes'] = strtotime(date("Y-m-d",$datum)."-6 day"); $kalendar['vege'] = strtotime(date("Y-m-d",$datum));}
65 }
66
67 /* Naptár segéd */
68 $kezdes = date("Y-m-d",$kalendar['kezdes']);
69 $vege = date("Y-m-d",$kalendar['vege']);
70
71 $selozo_het = strtotime(date("Y-m-d",$kalendar['kezdes'])."-1 week");
72 $kovetkezo_het = strtotime(date("Y-m-d",$kalendar['kezdes'])."+1 week");
73

```

12. ábra a naptár működése

Lekérdezzük az aktuális dátumot a \$datum változóba, ha a címsorban megtalálható, akkor azt használjuk, ellenkező esetben előállítjuk a time() függvénnyel (12. ábra). Az aktuális dátumot feldaraboljuk a date() függvény paraméterezésével év, hónap, hét, nap-ra külön változóknak tárolva (\$sev, \$honap, \$het, \$snap). A \$snap változó értéke 1-től 7-ig terjed, ahol az '1' a hétfőt a '7' pedig a vasárnapot jeleni értelem szerűen a köztes számok a köztes napokat jelentik egyező sorrendben. A \$kalendar tömbben tárolom majd az aktuális héthez tartozó adatokat. A hétfőválasztó algoritmus az aktuális nap értékéből (1-7) és az aktuális dátumból tudja kiszámolni, hogy milyen nap van ma (hétfő - vasárnap) és a mai naphoz viszonyítva melyik dátum volt a hét első napja, eleje (hétfő, ami a \$snap == 1) és melyik dátum ennek a hétnek a vége, utolsó napja (vasárnap, ami a \$snap==7). Tegyük fel, hogy ma csütörtök van, ami a hétnek a negyedik napja.

Az egymásba ágyazott *if* feltételeim közül az fog teljesülni, ahol a $\$nap=4$. Ha a negyedik napja van a hétnek, akkor tudjuk, hogy a hét első napja az a mai dátumhoz képest három nappal korábban volt, így a $\$kalendar$ tömbbőm kezdés ($\$kalendar['kezdes']$) értéke az aktuális dátum -3 nap lesz ($strtotime(date("Y-m-d",\$datum)."-3 day")$) és a hét utolsó napja három nap múlva lesz, így a $\$kalendar$ tömbbőm vég ($\$kalendar['vege']$) értéke az aktuális dátum +3 nap lesz ($strtotime(date("Y-m-d",\$datum)."+3 day")$). Ezek alapján az aktuális dátum napjához viszonyítva kiszámolhatjuk, hogy ennek a hétnek a hétfője és a vasárnapja hanyadikára esik. Ez lesz az alapja a naptáramnak, mivel mindig az aktuális hetet jeleníti meg, de a jövőben is bármilyen dátumból előállítja az ahhoz a naphoz tartozó hétfőt és vasárnapot is. A hetek lapozására két változót vettem fel $\$lozo_het$ és $\$kovetkezo_het$. Az $\$lozo_het$ változóban a $\$kalendar$ tömbb kezdés értékéből kivontam egy teljes hetet, míg a $\$kovetkezo_het$ változójához hozzáadtam egy hetet. Itt nyeri értelmét az, hogy a PHP program elején vizsgálom, hogy az url sávban létezik-e dátum értékkel ellátva, mert, ha nem, akkor az aktuális időt használom, de ha léptetek egy hetet előre vagy hátra, akkor azt az url sávban leteszem, így az oldal betöltődésekor azzal a dátummal fog dolgozni a PHP programom.

```
78 $naptar_nap["hetfo"] = date("Y-m-d",\$kalendar['kezdes']);
79 $naptar_nap["kedd"] = date("Y-m-d",strtotime(date("Y-m-d",\$kalendar['kezdes'])."+1 day"));
80 $naptar_nap["szerda"] = date("Y-m-d",strtotime(date("Y-m-d",\$kalendar['kezdes'])."+2 day"));
81 $naptar_nap["csutortok"] = date("Y-m-d",strtotime(date("Y-m-d",\$kalendar['kezdes'])."+3 day"));
82 $naptar_nap["pentek"] = date("Y-m-d",strtotime(date("Y-m-d",\$kalendar['kezdes'])."+4 day"));
83 $naptar_nap["szombat"] = date("Y-m-d",strtotime(date("Y-m-d",\$kalendar['kezdes'])."+5 day"));
84 $naptar_nap["vasarnap"] = date("Y-m-d",strtotime(date("Y-m-d",\$kalendar['kezdes'])."+6 day"));
85
```

13. ábra a hét napjainak tárolása

A naptár napjait (neveit) egy tömbben eltároltam. Tehát minden hétnek a kezdő napja hétfő ($\$naptar_nap["hetfo"] = date("Y-m-d",\$kalendar['kezdes']);$), minden kedd a hét kezdőnapjához egy nappal később van, minden szerda a hét kezdőnapjához két nappal később van és így tovább (13. ábra).


```

echo'
<h2>Időpontfoglalás</h2>
<p>Szöveg</p>
<p><a href="http://qualityworld.hu/php/booking.php?datum='.$selozo_het.'"> '$kezdes.'</a> - <a href="http://qualityworld.hu/php/booking.php?datum='.$skovetkezo_het.'"> '$vege.'</a>
<table class="table">
<thead>
<tr>
<th>Hétfő<br>'. $naptar_nap["hetfo"].'; if($user['admin']==1){echo'<br><a href="http://qualityworld.hu/php/admin_booking_upload.php?datum='.$naptar_date["hetfo"].'>Feltöltés</a>';}
<th>Kedd<br>'. $naptar_nap["kedd"].'; if($user['admin']==1){echo'<br><a href="http://qualityworld.hu/php/admin_booking_upload.php?datum='.$naptar_date["kedd"].'>Feltöltés</a>';}
<th>Szerda<br>'. $naptar_nap["szerda"].'; if($user['admin']==1){echo'<br><a href="http://qualityworld.hu/php/admin_booking_upload.php?datum='.$naptar_date["szerda"].'>Feltöltés</a>';}
<th>Csütörtök<br>'. $naptar_nap["csutortok"].'; if($user['admin']==1){echo'<br><a href="http://qualityworld.hu/php/admin_booking_upload.php?datum='.$naptar_date["csutortok"].'>Feltöltés</a>';}
<th>Péntek<br>'. $naptar_nap["pentek"].'; if($user['admin']==1){echo'<br><a href="http://qualityworld.hu/php/admin_booking_upload.php?datum='.$naptar_date["pentek"].'>Feltöltés</a>';}
<th>Szombat<br>'. $naptar_nap["szombat"].'; if($user['admin']==1){echo'<br><a href="http://qualityworld.hu/php/admin_booking_upload.php?datum='.$naptar_date["szombat"].'>Feltöltés</a>';}
<th>Vasárnap<br>'. $naptar_nap["vasarnap"].'; if($user['admin']==1){echo'<br><a href="http://qualityworld.hu/php/admin_booking_upload.php?datum='.$naptar_date["vasarnap"].'>Feltöltés</a>';}
</tr>
</thead>
<tbody>
<tr><td><table class="table table-hover"><tbody>
';

```

14. ábra a naptár fejléce

A fentieket felhasználva iratom ki a PHP *echo* '' funkciójával az időpontfoglaló naptáramat. A két lapozó gombomat, ami a hét első és utolsó dátumaként jelenik meg (*\$kezdes*, *\$vege*) és az url sávban a dátum értéket változtatva irányítja a naptárat (*\$selozo_het*, *\$kovetkezo_het*). A fejléchez (14. ábra) a *\$naptari_nap* tömb használatával iratom ki a dátumokat és minden egyes napnál van egy vizsgálat, hogy admin-e a felhasználó, mert, ha admin, akkor megjelenik a feltöltés gomb, ami az *admin_booking_upload.php* fájlra hivatkozik annak átadva a fejlécen megjelenő kiválasztott nap dátumát, ezzel megjelenik az időpontok feltöltése felület az adott naphoz.

```

//Hétfői időpontok kilistázása
$q = mysqli_query($conn,"SELECT * FROM booking WHERE b_date > ".$naptar_date["hetfo"]." AND b_date < ".$naptar_date["hetfo"] + 86400."");
while($q_r=mysqli_fetch_assoc($q))
{
    if($q_r['uid']>0)
    {
        echo ' <tr><td><del>'.date("H:i",$q_r['b_date']).' - '.date("H:i",strtotime(date("H:i",$q_r['b_date'])."+1 hour")).'</del></td></tr>';
    }else{
        echo ' <tr><td><a href="http://qualityworld.hu/php/booking_details.php?id='.$q_r['id'].'">'.date("H:i",$q_r['b_date']).' - '.date("H:i",strtotime(date("H:i",$q_r['b_date'])."+1 hour")).'</a></td></tr>';
    }
}
echo ' </tbody></table> </td><td><table class="table table-hover"><tbody>';

```

15. ábra az időpontok kilistázása

Minden naphoz tartozik egy kilistázás (15. ábra), SQL lekérdezés, ami lekérdezi a *booking* táblából azokat a rekordokat, ahol a *b_date* nagyobb, mint a *\$naptar_date*-ben tárolt például hétfői nap dátuma, de kisebb, mint a hétfői nap dátuma + 24 óra. Ha egy időpont le van foglalva, azt onnan tudjuk, hogy a *booking* táblában az adott rekord uid mezője nem nulla, így ezt vizsgáljuk.

Áthúzva jelenítjük meg a foglalt időpontokat és ezekre kattintva nem történik semmi, de ha a rekord uid-ja nem nagyobb, mint nulla, tehát szabad, akkor meghívjuk a *booking_details.php* programot és átadjuk neki az id-t, ezzel megjelenik a kiválasztott időpont lefoglalására a felület.

```
//Saját foglalások megjelenítése
$q = mysqli_query($conn,"SELECT * FROM booking WHERE complete = 0 AND uid = ".$_COOKIE['uid']);
$van=false;
while($q_r=mysqli_fetch_assoc($q))
{
    $van=true;
    echo '<tr><td>'.date("Y-m-d H:i",$q_r['b_date']).' - '.date("H:i",strtotime(date("Y-m-d H:i",$q_r['b_date'])."+1 hour")).' <a href="http://qualityworld.h
```

16. ábra saját foglalásaim

Minden felhasználó számára a naptár alatt kilistázom az általuk foglalt időpontokat (16. ábra). Ehhez SQL queryben vizsgálom, hogy a booking táblából azokat a rekordokat listázza ki, ahol a complete flag egyenlő '0'-val, tehát még nem teljesült a foglalás és ahol az uid egyenlő a COOKIE-ban tárolt 'uid'-val, így minden felhasználónak csak a saját foglalásai jelennek meg.

Időpont lefoglalása – booking_details.php

Amikor általános felhasználóként lefoglalunk egy időpontot, akkor a *booking_details* PHP program fogja feltölteni az adatbázisba a foglalásunkat. Sütikből beazonosítja a felhasználónkat, *\$user* tömbbe kimenti a többi adatát a user rekordunknak, a *booking.php*-től kapott *booking id*-t GET metódussal átveszünk és a foglalás adatait kimentjük a *\$booking* tömbbe. Amikor kitöltöttük a megjegyzés mezőt és rákattintottunk az időpontfoglalás gombra, akkor egy UPDATE SQL paranccsal beállítjuk a *booking* táblában az időpont rekordjának az *uid*-ját a COOKIE-ban található *user id*-ra, a *notes* mezőhöz hozzáadjuk a megjegyzésbe beírt szöveget miután megtisztítottuk a *mysqli_real_escape_string()* függvényvel. Az időpontfoglalásról az alkalmazás rendszerüzenetet küld a felhasználónak és a cég közös emailcímére is, hogy minden érintett értesüljön a sikeres foglalásról.

Időpontok kezelése

Minden felhasználó a foglaláson felül tudja **törölni** az általa lefoglalt időpontokat (felszabadítani). Ezt a *booking_del.php* program végzi el. A GET metódussal kapott *id* alapján beazonosítja az időpontot, majd annak az *uid* flag-jét átállítja nullára, szabadon foglalhatóvá téve ezzel a beazonosított időpontot. Ennek hatására nem csak a naptárban jelenik meg újra foglalhatóként, de a saját foglalásaink közül is eltűnik az időpont. Erről email értesítést kap a cég, egy közös emailcímére.

Az adminisztrátorok **fel tudnak tölteni** a naptár fejlécén szereplő feltöltés gombokkal bármely naphoz időpontokat. Ezt az *admin_booking_upload* PHP program meghívásával végzi az alkalmazás.

```
32 //POST FELDOLGOZÁS
33 if(isset($_POST['mentes']))
34 {
35     $datum = $_GET['datum'] + $_POST['ora']*60*60;
36
37     for($i=1;$i<=$_POST['db'];$i++)
38     {
39         $id = mysqli_query($conn,"INSERT INTO booking (c_date,m_date,b_date) VALUES ('".time()."','".time()."','".$datum."')");
40         $datum+=(60*60);
41     }
42
43     header('Location: http://qualityworld.hu/php/booking.php');
44 }
45
```

17. ábra időpontok feltöltése - admin funkció

Az url sávból kinyeri az aktuális dátumot GET metódussal, majd a form-on bevitt kezdő időpontot (*ora*) megkapja POST metódussal a PHP program. Ennek a két értéknek az összegét tárolom a *\$datum* változóban, majd egy *for* ciklussal, ami annyiszor fut, ahány darab időpontot kért az adminisztrátor (*\$_POST['db']*). A cikluson belül INSERT query-vel felviszem a *booking* táblába először a *\$datum* kezdőértéket a *b_date* (booking date) mezőbe, majd a *c_date*, *m_date* (create date, modify date) mezőkbe az aktuális időt (*time()*). Minden ciklusban hozzáadok a *\$datum* változóhoz egy órát (17. ábra). Később itt lehetne változtatni az időpontok hosszán.

```

29 if(!empty($user) && $user['admin']==1)
30 {
31     $q = mysqli_query($conn,"SELECT * FROM booking WHERE id = '". $_GET['id']."'");
32
33     $booking = array();
34     while($q_r=mysqli_fetch_assoc($q))
35     {
36         $booking = $q_r;
37     }
38
39 if(!empty($booking))
40 {
41     mysqli_query($conn,"DELETE FROM booking WHERE id = '".mysqli_real_escape_string($conn,$_GET['id'])."'");
42
43     //Email-13*/

```

18. ábra időpontok törlése - admin funkció

Amikor a felvitt időpontok még szabadok, azokat az időpontokat az adminisztrátorok **törölhetik** (18. ábra). Ekkor az *admin_booking_del.php* program fut le. Itt miután beazonosítottuk a felhasználót (*\$_COOKIE['uid']*) még meg kell vizsgálnunk, hogy adminisztrátor-e (*\$user['admin']==1*) és ha az, akkor egy DELETE queryt indít el, ami a *booking* táblából törli azt a rekordot, aminek az *id*-jét megkaptuk az url-ből. Az időpont törléséről email értesítést kap a cég egy közös emailcímén keresztül.

Ha már egy időpontot lefoglalt valaki, akkor az nem törölhető egy lépésből, először **fel kell szabadítani**, amit az *admin_booking_clear* PHP program hajt végre. Ez a booking táblában az adott időpont rekordján a *uid*-t beállítja nullára és amelyik időponthoz nincs hozzárendelve user id, az szabad. Erről is megy email értesítés.

A foglalt időpontokat **le lehet zárni** a munka elvégzésével. Az *admin_booking_close.php* programmal a *booking* táblában, az adott időpont rekordján a *complete* flag-et '1'-re állítja. Ennek hatására többet nem szerkeszthető az adott időpont és hozzá tartozó opciói. Email rendszerüzenetet küld a lezárásról is a program.

Hozzá lehet rendelni szerelőket minden egyes foglalt időponthoz. Az *admin_booking_workers.php* végzi ezt a lépést. A foglalt időpont melletti szerelők gombra kattintva megjelennek a vég aktuális szerelői, akik közül többet is hozzá tudunk rendelni egy időponthoz a checkbox-ok bepipálásával. Akiket bepipáltunk, a szerelő rögzítése gomb hatására beíródnak a *booking_workers* táblába, ahol a *bid* maga a booking id (*\$booking['id']*), ez a foglalás azonosítója és mellé a *wid*-be, ami a worker id és a *c_date*-be (create date) az aktuális dátum. Ez az INSERT query annyiszor ismétlődik, ahány szerelőt hozzá szeretnénk rendelni az adott foglaláshoz.

Mivel a *booking_workers* tábla egy kapcsolótábla így egy foglalás többször is szerepelhet a táblában különböző szerelőkkel és egy szerelő is szerepelhet benne többször, különböző foglalásokhoz rendelve. Az időpontokhoz hozzárendelt szerelők megjelennek a foglalások alatt az admin felületen, ha ezen szeretnénk módosítani, akkor újra megnyitva a szerelők opciót kivethetjük a pipát bármely szerelő neve mellől és azok a *booking* és *worker* kapcsolatok (rekordok) törlésre kerülnek a *booking_workes* táblából.

Szerelők felvételéhez át kell lépnünk az Admin – Szerelők kezelése menüpontra, ami az *admin_workers.php* fájlt nyitja meg. Felül kilistázza a jelenlegi szerelőket a *workers* táblából. Mindegyik szerelőnél van egy törlés gomb, ez meghívja az *admin_workers_del.php* fájlt, ami egy DELETE query-t futtat a *workers* táblán, ahol a dolgozó *id*-ja egyezik a kapott *id*-val. Ha pedig egy új szerelőt szeretnénk rögzíteni, akkor a *workers* táblába feltölti a formon kitöltött név, emailcím, telefonszám mezőt (*name*, *email*, *phone*) és az aktuális dátumot a *c_date* mezőbe. Sikeres feltöltés után megjelenik a rögzített szerelőknél, hozzárendelhetővé válik az összes foglalt rendelésnél és emailben rendszerüzenet is küld róla az alkalmazás.

3.3.3 Biztonságtechnikai szempontok

Az általam ismert és elfogadott alapvető biztonságtechnikai szempontoknak megfelelően fejlesztettem, odafigyeltem a program tervezése és fejlesztése közben. Ez jelen van már a webtárhely szolgáltatás cPanel bejelentkezésétől kezdve a mezei regisztrált felhasználó profiljáig. A cPanel felületen tudom létrehozni és elérni is az adatbázisokat, adatbázis felhasználóit, e-mail fiókokat, ftp kapcsolatokat és társait így mindenképp erős jelszóval kellett ellátni azt a profilt. Ezeknek a jelszókövetelményeknek, mint például a karakterek minimum száma, speciális karakterek, kisbetű, nagybetű, számok, értelmetlen, egyedi sorozatának megfelel minden egyes általam alkalmazott hozzáférés jelszava. A látogatók által készült regisztrációk jelszavai is titkosítva kerülnek tárolásra az adatbázisban az md5 függvény használatával, ezen felül több ötletem is felmerült. Például az md5 függvényt is be lehet ágyazni még egy md5 függvénybe, sha1, vagy hasonló kódolási függvény a base64, amivel titkosítva tárolhatjuk a jelszavakat. Esetleg a salt, amivel a jelszó elejére, végére vagy éppen középre hozzáilleszthetünk egy tetszőleges karaktersorozatot és azzal együtt titkosítjuk. Továbbá a MySQL felhasználó is, amit használ a rendszerem csak a megfelelő jogosultságokkal rendelkezik. Nem tud táblákat létrehozni vagy törölni, amelyik táblán csak

ellenőrizni kell, ott csak read jogosultsága van. Természetesen odafigyeltem a modify, delete, write jogokra is az esetleges támadások elkerülése végett. A felhasználók által kitöltött beviteli mezők értékeit a *mysqli_real_escape_string()* függvénnyel tisztítottam meg, hogy ne tudják módosítani az SQL lekérdezéseim működését, ne ágyazhassanak bele kártevő kódot, esetleg jussanak hozzá illetéktelenül plusz információkhoz, röviden, védelmet biztosít az SQL injection típusú támadások ellen. Összesítve a biztonságos jelszavak és akár a kevésbé biztonságos jelszavak is, de megfelelő titkosítás mellett vannak tárolva. Megfelelnek az általam fejlesztett rendszerben tárolt adatok megvédésére és emellett a jogosultságok kezelése és aprólékos szétdarabolása is megvéd a rosszindulatú támadásoktól.

4 Tesztelés

Ebben a részben ismertetem a webalkalmazásom funkcionak és egész működésének a teszteléseit. Fejlesztés közben minden egyes modul elkészülte közben és utána is leellenőriztem működését és hogy a specifikációknak is megfeleljen.

Főoldal reszponzivitásának és beépített moduljainak tesztelése. Indítottam a saját számítógépen egy böngészőt, ahol betöltöttem a weboldal főoldalát. Elsőként a navigációs sáv menüpontjait végig próbáltam, odairányít-e, ahova szeretném és végig a képernyő tetején marad-e a *navbar*. Az ajánlatkérő gomb átirányít a bejelentkező felületre. A fejléc alatt megjelenő slideshow automatikusan lapoz, ha felette hagytam a kurzort, addig megáll, az indikátorok jelzik a diavetítés státuszát és jobbra-balra navigáció is a kívánt eredményt produkálja. A lentebb felsorolt szövegdobozok és képek a megfelelő helyen és megfelelő formában jelennek meg, nem fedik le egymást. A kapcsolat gomb odanavigál a kapcsolat szekcióhoz. Itt a név, emailcím és üzenet kitöltése után el tudjuk küldeni az üzenetet, amiről azonnali értesítést kap az érdeklődő is a megadott email címen és a cég munkatársai is közös mailbox-ukban. A mailto hivatkozás a kapcsolat bal hasábjában elindítja az alapértelmezett levelezőmet és a megfelelő címmel indít új levelet. A láblécben szereplő hivatkozás a qualityworld.hu-ra a főoldalra irányít, a felette található nyíl pedig az oldal tetejére ugrik. Ezeket a teszteléseket végig próbáltam összébb húzott böngészőablakban is, illetve mobil eszközökön is, ezzel is az oldal reszponzivitását vizsgálva. Ehhez a ponthoz kiegészítésképpen hozzáfűzném, hogy a navigációs sáv is a megfelelő pillanatban

(kijelzőszélesség) csoportosítja a menüpontokat és annak menü ikonjára kattintva lenyílik a menüsor és elérhetővé válnak a menüpontok.

Regisztráció. A regisztrációs felület kitöltésekor leteszteltem azokat az eseteket, amikor az input mezők üresen, vagy pedig nem megfelelően vannak kitöltve, ezekben az esetekben hibával nem engedett regisztrálni a rendszer. A beépített recaptcha is az elvártak szerint működik, csak akkor enged regisztrálni, amikor be van pipálva a nem vagyok robot checkbox, és az input mezők megfelelően ki vannak töltve. A regisztráció után megjelenik a qualityw_online adatbázisában a felhasználó rekordja, de még inaktív státuszban. A felhasználó a megadott email címre kapott megerősítő levelet, amiben a generált hash kódot megfelelően tartalmazza a hivatkozás, erre kattintva aktívvá is válik a felhasználó hozzáférése. Abban az esetben, ha ugyan azzal az email címmel szeretnék újra regisztrálni, hibát dob a rendszer, mivel már létezik ilyen levelezési címmel felhasználói rekord az adatbázisban.

Bejelentkezés. Az email cím és jelszó mezők kitöltése továbbra is feltétele a bejelentkezés sikerességének. Hiányos és vagy nem megfelelő formátum esetén a javascript nem engedni elküldeni sem a bejelentkezési kérelmet. Sikeres bejelentkezés esetén létrehozza a sütit az alkalmazás, az 1 órás érvényességi idővel, majd elérhetővé válik az időpontfoglaló oldal.

Időpontfoglalás. A naptárban az aktuális hét fogad minket, ha léptetjük előre-hátra a hetet, akkor a kívánt napok fogadnak a naptárban. Az url sávban megjelennek az éppen aktuális hét kezdődátumai léptetés hatására (...*booking.php?datum=1544396400*). A naptár időpontjainak válogatása közben a kurzor elhaladására (*hover*) jól láthatóan kiemelkedik az adott időpont, rákattintva pedig a megfelelő nap lefoglalására navigál. Sikeres foglalás esetén megérkezik a rendszerüzenet, majd megjelenik a felhasználó foglalt időpontja között a rekord és az admin nézetben az összesített foglalások oszlopában is (természetesen az adatbázis *bookings* táblájából jelenítik meg ezek a nézetek). Egy foglalt időpontot még egyszer nem tudunk lefoglalni, csak, ha töröljük a saját foglalásaink közül, esetleg az adminisztrátor az összesített foglalások nézetben fel nem szabadítja. Ennek hatására jelenik meg, válik újra foglalhatóvá az időpont. A foglalás törléséről rendszerüzenet érkezik. Kijelentkezés esetén újra a bejelentkező felületre kerülünk, a böngészőből eltűnik a süti is.

Admin funkciók. A naptár fejlécében szereplő feltöltés hivatkozások hatására a kívánt feltöltő felület jelenik meg. Itt próbaként egy üres szombati napot az alapértelmezett értékekkel, reggel 8-tól kezdve 8 db egy órás időponttal töltöttem fel a naptárat, melyek megjelentek a heti nézetben foglalható időpontokként és a booking táblába is felvételre kerültek. Erről egyből érkezett is a rendszerüzenet, hogy új időpontok lettek felvéve. Tovább lépve a foglalt időpontok admin nézetre, a szabad időpontok megjelennek üres-ként. Ezeket a törlés hivatkozással el tudjuk távolítani, melyről email értesítés érkezik és az adatbázisból is törlésre kerül az adott rekord. A foglalt időpontok esetén a további gombok elérhetőek és a bővebb leírás. Felszabadításra kattintva a foglalás lekerül az időpontról. A szerelőkre kattintva hozzárendelünk a felvitt dolgozók közül egyet a foglaláshoz, ez megjelenik a leírásban és újra a szerelők menübe lépve bepipálva marad a választásunk. Több szerelő felvétele esetén több rekord kerül be a *booking_workers* táblába, ha pedig kivesszük a checkbox pipáját, akkor törlésre kerül a táblából azon a booking id-n az a worker id-jú rekord. A foglalás lezárásakor elszürkül a foglalás és teljesítettként jelenik meg.

A tesztelést több eszközzel, több népszerű böngészővel és több ablakméretben hajtottam végre. A böngészők közül a legfrissebb Google Chrome-ot, Internet Explorer-t, Microsoft Edge-t és Mozilla Firefox-ot használtam és ezeknek a mobil verzióit. A felsorolt böngészők mindegyikében kifogástalanul működtek a web alkalmazásom funkciói és eleget tettek a responzív megjelenésnek.

Eszköz típusa	Képernyőfelbontása	Képernyő mérete (inch)	Tesztelt böngészők
Lenovo ThinkPad T480	1920x1080	14"	Google Chrome, Internet Explorer, Microsoft Edge, Mozilla Firefox
Acer TimelineX 5830TG	1377x768	15.6"	
iPad 2018 6.gen	2048x1536	9.7"	Safari
Motorola Moto Z	2560x1440	5.5"	Google Chrome mobile

5 Összegzés

Felkérést kapam a QualityWorld Kft.-től, hogy szeretnének egy bemutatkozó weboldalt, ami rendelkezik egy időpontfoglaló modullal. Egyik fő szempont az volt, hogy a webalkalmazás legyen reszponzív megjelenésű, így nem csak otthonról számítógépen, hanem mobilról is használható. Ennek megvalósítását HTML [7], CSS [6] nyelvekkel és Bootstrap [3] keretrendszer integrálásával hajtottam végre. A weboldal másik fő feladata az online időpontfoglalás megvalósítása. Ehhez kellett készítenem a felhasználók bejelentkezéséhez és regisztrációjához egy PHP [1] nyelvű felületet, ami kommunikál a MySQL [5] adatbázissal, hogy létre tudjunk hozni új felhasználókat és a meglévő userek adatait össze tudjuk hasonlítani bejelentkezéskor megadott értékekkel. Az alkalmazásban nyilvántartjuk a cégnél dolgozó szerelőket. Egyedi időpontfoglaló rendszer készült, ahova az adminisztrátorok dinamikusan tudnak időpontokat felvenni és ezeket az oldal látogatói előzetes regisztráció után le is tudják foglalni a saját nevükre. Ez nagy segítség a cég dolgozói számára, mivel sokkal gyorsabban, könnyebben működik az időpont egyeztetése a megrendelőkkel és egyszerűbb a kapcsolatfelvétel is. Nem maradnak elfeledett rendelések és minden módosításról email értesítést kapnak.

A webalkalmazás tervezése majd fejlesztése során meg kellett ismerkednem a webprogramozásban rejlő lehetőségekkel, a HTML5, PHP5, CSS3, MySQL5 technológiák és programozás nyelvek sajátosságaival. Először a főoldal megtervezésével kezdtem, közben már gondolkodtam az időpontfoglalás és az azt körülölelő. Az adatbázis tervezésénél törekedtem az optimalizáltságra, hogy ne készítssek felesleges mezőket, emellett a jövőre gondolva legyen elég mező és tábla a könnyű továbbfejleszthetőség miatt. A PHP kódok írásakor is ügyeltem a továbbfejleszthetőségre, próbáltam logikusan felépíteni a web alkalmazásnak a működését. Figyeltem, hogy az adatokat biztonságosan kezeljem és tároljam az adatbázisban. Itt több biztonságtechnikai szempontot is figyelembe vettem a fejlesztés során, amikkel szemben támasztott elképzelésemnek megfelel a kész alkalmazás. A közeljövőben kerül promótálásra az alkalmazás. Én még rengeteg lehetőséget látok ebben az irányban, de most a cégen és a vásárlói körén múlik, hogy milyen szükséges funkciók kerüljenek még beépítésre és mi az, amit esetleg át kell alakítani. A jövőben még fejleszteni, bővíteni szeretném és fontos cél, hogy riportokat lehessen generálni az összegyűlt adatokból. Remélem, hogy az alkalmazásom elősegíti a vállalkozás további sikeres működését.

6 Summary

I have been asked by QualityWorld Ltd. that they would like to introduce a website with an online booking module. One of the main aspects was that the web application should be responsive, so it is not only possible to display the full length of the website from home on computer, but also from mobile on the go, simply reserving time for anyone and anywhere. It was implemented with HTML, CSS languages and the integration of the Bootstrap framework. Another major task of the website is to implement the online booking. To do this, I had to create an interface for users to login and sign up, which communicates and modifies them in the MySQL database in PHP programming language. So that we can create new users and compare existing user data with values entered at login. In the application, we keep track of the contractors working in the company. A unique time reserving system has been developed, where administrators can dynamically upload dates, and the visitors can book them with their names on pre-registration. This is a great help for the company's employees, since much more quickly, it is easier to match the time with the customers and make the contact easier. There will be no forgotten orders and every change will be notified via email.

During the design and the development of the web application, I had to learn the features of web programming, the features of HTML5, PHP5, CSS3 and MySQL5 programming languages. First, I started designing the start page, while I was thinking about the booking and the features, what to do and how it will work, the whole thing in complex. When I am designing a database, I tried to optimize it so that I do not create unnecessary fields and think about the future to make enough field and table can be easy to upgrade. Even when writing PHP codes, I kept upgrading, trying to logically build the complex web application. When working with data, I paid attention not only to generating, but also to storing them securely in the database. Here, I have observed several aspects of security in the development process, which I have come up with in my application. The application will be promoted in the near future. I still have a lot of opportunities in this direction, but now it is up to the company and its customers to see what features are needed to be built in and what needs to be changed. In the future, I would like to continue the development and have a great desire to be able to extract reports from the collected data. Or get better in the details. I hope they will be positive about my application.

7 Irodalomjegyzék

[1] Sági Gábor: Webes adatbázis-kezelés MYSQL és PHP használatával, 2005
ISBN: 978-963-9425-04-0

[2] George Schlossnagle: PHP fejlesztés felsőfokon, 2004, ISBN: 963-9301-80-9

[3] Bootstrap: <https://getbootstrap.com/>
Érvényességi dátum: 2018.11.19.

[4] PHP: <https://www.w3schools.com/php/default.asp>
Érvényességi dátum: 2018.11.23.

[5] Kozmajer Viktor: PHP és MySQL az alapoktól, 2011, ISBN: 978-963-9425-74-3

[6] Virginia DeBolt: HTML és CSS - Webszerkesztés stílusosan, 2005
ISBN: 978-963-9301-96-2

[7] Bruce Lawson - Remy Sharp: Bemutatkozik a HTML 5, 2013
ISBN: 978-963-9929-28-9

8 Ábrajegyzék

1. ábra Relációs adatmodell, a Qualityw_online adatbázi ER diagrammja.....	15
2. ábra Navigációs sáv működése.....	22
3. ábra Slideshow működése	22
4. ábra kapcsolatfelvételi modul felépítése.....	24
5. ábra bejelentkezés osztály	25
6. ábra kijelentkezés funkció	27
7. ábra Google reCAPTCHA HTML beépítése	27
8. ábra Google reCAPTCHA szerver oldali php működése.....	28
9. ábra regisztráció logikája.....	29
10. ábra regisztráció megerősítésének működése.....	30
11. ábra a felhasználó adatai sütiből.....	31
12. ábra a naptár működése	31
13. ábra a hét napjainak tárolása.....	32
14. ábra a naptár fejléce	33
15. ábra az időpontok kilistázása.....	33
16. ábra saját foglalásaim	34
17. ábra időpontok feltöltése - admin funkció.....	35
18. ábra időpontok törlése - admin funkció.....	36