

1. BEVEZETÉS

A Sports Resource Booking alkalmazást azért fejlesztették ki, hogy lehetővé tegye a diákoknak foglaljon sportforrásokat. A rendszergazda nyomon követheti az erőforrások elérhetőségét a finom eljárás egyszerűsítésével együtt. Ez az alkalmazás legyőzi a bennszülött problémákat a meglévő kézi rendszer. Ezenkívül rendkívül egyszerűvé teszi a hallgatók számára az erőforrások lefoglalását vagy ellenőrizheti az erőforrások elérhetőségét, megtekintheti az esetleges díjakat, valamint a foglalások előzményeit és az adminisztrátor számára források kiadása, pénzbírságok kiszabása és a foglalások előzményeinek megtekintése. Az API Az ehhez az alkalmazáshoz kifejlesztett alkalmazás a Heroku-n van telepítve, és készen áll a kérések megválaszolására az alkalmazás által az adatbázis elérésével várt kimenetekkel.

1.1 A probléma meghatározása

A projekt két fő modulból áll:

FELHASZNÁLÓ

- Hallgatói bejelentkezés
- A kezdőlapon megjelenik az összes rendelkezésre álló erőforrás és forrás, amelyet lefoglalt
- Foglaljon le egy forrást, amely csak 20 percre lesz fenntartva neki
- Foglалás lemondása
- A foglalások története
- Csak 1 foglalásra korlátozódik
- Az erőforrások 12:00 és 16:00 óra között adhatók ki, így a foglalás órától indulhat 11:40 és 15:00 között – A letiltott felhasználók nem foglalhatnak le erőforrásokat, amíg ki nem fizetik a esedékes.

ADMIN

- Admin bejelentkezés
- Adjon hozzá erőforrásokat
- Erőforrások törlése
- (A rendelkezésre álló erőforrások foglalási csökkentését a háttérben kell elvégezni)
- Erőforrások kiadása és a foglalás frissítése

- (Minden erőforrást vissza kell küldeni 16:20-ig, különben bírság kerül felszámításra és blokkolásra automatikusan)
- Frissítse a foglalást, miután a felhasználó visszaküldi az erőforrásokat.
- Fizikailag finom kifizetéseket kaphat, és feloldhatja a felhasználó blokkolását

1.2 A projekt hatóköre

Ez az alkalmazás használható a kézi rendszerrel kapcsolatos problémák csökkentésére.

Az adminisztrátor egy kattintással kiadhatja és visszautasíthatja a foglalást, és még a tartásban is segít a Foglalási előzmények és a felhasználók letiltott listájának füle .Az adminisztrátor hozzáadhat vagy törölje az erőforrásokat .Ez megkönnyíti a rendszergazda számára az összes erőforrás nyomon követését és foglalások. A mobilalkalmazás nagyon egyszerűvé teszi a felhasználók számára az erőforrások lefoglalását és az ellenőrzést korábbi foglalásaik története.

2. TECHNOLÓGIA STACK

2.1 Kezelőfelület kialakítás:

2.1.1 Webes alkalmazás: HTML, CSS, BOOTSTRAP, JavaScript

A Hypertext Markup Language (HTML) a szabványos jelölőnyelv weboldalak és webes alkalmazások létrehozásához. A Cascading Style Sheets (CSS) és a JavaScript segítségével a világháló sarokköve-technológiáinak hármását alkotja. A webböngészők HTML-dokumentumokat fogadnak webszerverről vagy helyi tárhelyről, és a dokumentumokat multimédiás weboldalakká jelenítik meg. A HTML szemantikailag írja le a weboldal szerkezetét, és eredetileg a dokumentum megjelenésére utaló jeleket tartalmazott.

A Cascading Style Sheets (CSS) egy stíluslapnyelv, amelyet a HTML-hez hasonló jelölőnyelven írt dokumentumok megjelenítésének leírására használnak. A CSS a World Wide Web egyik sarokköve, a HTML és a JavaScript mellett. A CSS-t úgy tervezték, hogy lehetővé tegye az elkülönítést a megjelenítés és a tartalom, beleértve az elrendezést, a színeket és a betűtípusokat. Ez a szétválasztás javíthatja a tartalom hozzáférhetőségét, nagyobb rugalmasságot és ellenőrzést biztosít a megjelenítési jellemzők meghatározásában, lehetővé teszi több weboldal számára a formázás megosztását a megfelelő CSS-nek egy külön .css fájlban történő megadásával, és csökkenti a szerkezeti tartalom bonyolultságát és ismétlődését.

A Bootstrap egy ingyenes és nyílt forráskódú előtér-könyvtár webhelyek és webes alkalmazások tervezésére. HTML- és CSS-alapú tervezősablonokat tartalmaz tipográfiához, űrlapokhoz, gombokhoz, navigációhoz és egyéb interfész-összetevőkhöz, valamint opcionális JavaScript-bővítményeket. Sok webes keretrendszertől eltérően ez csak a front-end fejlesztéssel foglalkozik.

2.1.2 Android alkalmazás: JAVA, XML

A Java egy magas szintű programozási nyelv, amelyet a Sun Microsystems fejlesztett ki. Eredetileg set-top boxokhoz és kézi eszközökhöz készült programok fejlesztésére tervezték, de később népszerű választás lett webes alkalmazások készítésére. A Java szintaxis hasonló a C++-hoz, de szigorúan egy objektum-orientált programozási nyelv. .Eltérően a Windows végrehajtható fájlaktól (.EXE fájlok) vagy a Macintosh alkalmazásoktól (.APP fájlok), a Java programokat nem közvetlenül az operációs rendszer futtatja. Ehelyett a Java programokat a Java Virtual Machine vagy JVM értelmezi, amely több platformon fut.

Az XML kód, a World Wide Web Consortium (W3C) hivatalos ajánlása, hasonló a Hypertext Markup Language (HTML) nyelvhez. Az XML és a HTML is tartalmaz jelölést

szimbólumok az oldal vagy fájl tartalmának leírására. Az XML-dokumentum alapvető építőeleme egy címkék által meghatározott elem. Egy elemnek van kezdő és záró címkéje. Az XML-dokumentum minden eleme egy legkülső elembe, az úgynevezett gyökérelembe található. Az XML támogatja a beágyazott elemeket vagy az elemeken belüli elemeket is. Ez a képesség lehetővé teszi az XML számára, hogy támogassa a hierarchikus struktúrákat. Az elemnevek az elem tartalmát írják le, a szerkezet pedig az elemek közötti kapcsolatot.

2.2 Háttér:

2.2.1 Webes alkalmazás: Django

A Django egy ingyenes és nyílt forráskódú webalkalmazás-keretrendszer, amely Python nyelven íródott.)A Django saját elnevezési rendszerrel rendelkezik minden funkcióhoz és összetevőhöz (pl. a HTTP-válaszok „nézeteknek” nevezzük). A Django keretrendszer a gyors fejlesztés alapelveit használja, ami azt jelenti, hogy a fejlesztők egyszerre több iterációt is végrehajthatnak anélkül, hogy a teljes ütemezést előlről kezdenék. A Django segítségével bármilyen méretű és kapacitású projektet megbirkózhat, legyen szó egy egyszerű webhelyről vagy nagy terhelésű webalkalmazás.

2.2.2 Adatbázis: MySQL

A MySQL egy Oracle által támogatott nyílt forráskódú relációs adatbázis-kezelő rendszer (RDBMS), amely strukturált lekérdezési nyelven alapul. A szerkezet lekérdezési nyelven (SQL) alapul, amelyet információk hozzáadására, eltávolítására és módosítására használnak az adatbázisban. A szabványos SQL-parancsok, például az ADD, DROP, INSERT és UPDATE használhatók a MySQL-lel.

2.2.3 RESTful API: Lombik

A Flask egy Python nyelven írt webalkalmazás-keretrendszer. A Flask a WSGI (Web Server Gateway Interface) eszközkészleten és a Jinja2 sablonmotoron alapul.)A WSGI alapvetően egy protokoll úgy van meghatározva, hogy a Python-alkalmazások kommunikálni tudjanak egy webszerverrel, és így a CGI-n kívül webalkalmazásként használhatók.

3.IRODALMI FELMÉRÉS

3.1 Bevezetés

A Sports Resources Management System egy webes és egy Android-alkalmazás. Az adminisztrátor hozzáadhat és törölhet erőforrásokat a weben, és kezelheti a felhasználókat az erőforrások elfogadásában és elutasításában. A felhasználók lefoglalhatják az erőforrásokat az Android alkalmazásból, és 20 percen belül átvehetik azokat az irodában, és akár törölhetik is a kérést.

3.2 Meglévő rendszer

A meglévő rendszerben a felhasználóknak manuálisan kell felkeresniük a sportirodát és összegyűjteniük az erőforrásokat. Valamint a felhasználók nem foglalhatnak le egy erőforrást, bárhol is tartózkodik. Bár a felhasználó le van tiltva, bemehet az irodába. Az adminisztrátornak kézzel kell ellenőriznie a felhasználót, hogy szerepel-e a blokkolt listán, és ki kell adnia az erőforrást.

3.3 Javasolt rendszer

A javasolt rendszerben a felhasználók bárhol lefoglalhatják és lefoglalhatják az erőforrást, és 20 percen belül átvehetik. Ebben a rendszerben, ha egy felhasználó blokkolva van, nem tudja lefoglalni az új erőforrást, amíg nem törli a bírságot. A felhasználó ellenőrizheti a bírságot, az általa fizetendő összeget az Android alkalmazásban.

3.4 Szoftverkövetelmény specifikáció

3.4.1 Felhasználói követelmények

Van egy admin és a felhasználó, azaz a diák. A felhasználó minimális követelménye, hogy értse az alapvető angol nyelvet, és tudja, hogyan kell forrást lefoglalni, ellenőrizni a bírságot és beszélni a sporthivatalból. Az adminisztrátornak tudnia kell, hogyan fogadhatja el a felhasználók kérését, blokkolhatja a felhasználót, hogyan bocsáthat ki forrásokat manuálisan, és hogyan szedheti be a bírságokat a felhasználóktól, és hogyan oldhatja fel a felhasználók blokkolását.

3.4.2 Szoftverkövetelmény

-)Bármilyen webböngésző
-)Python 3 Django modullal
- 64 bites operációs rendszer.
- Emulátor
- Android stúdió
- csipog
- virtualenv
- Virtualenvwrapper

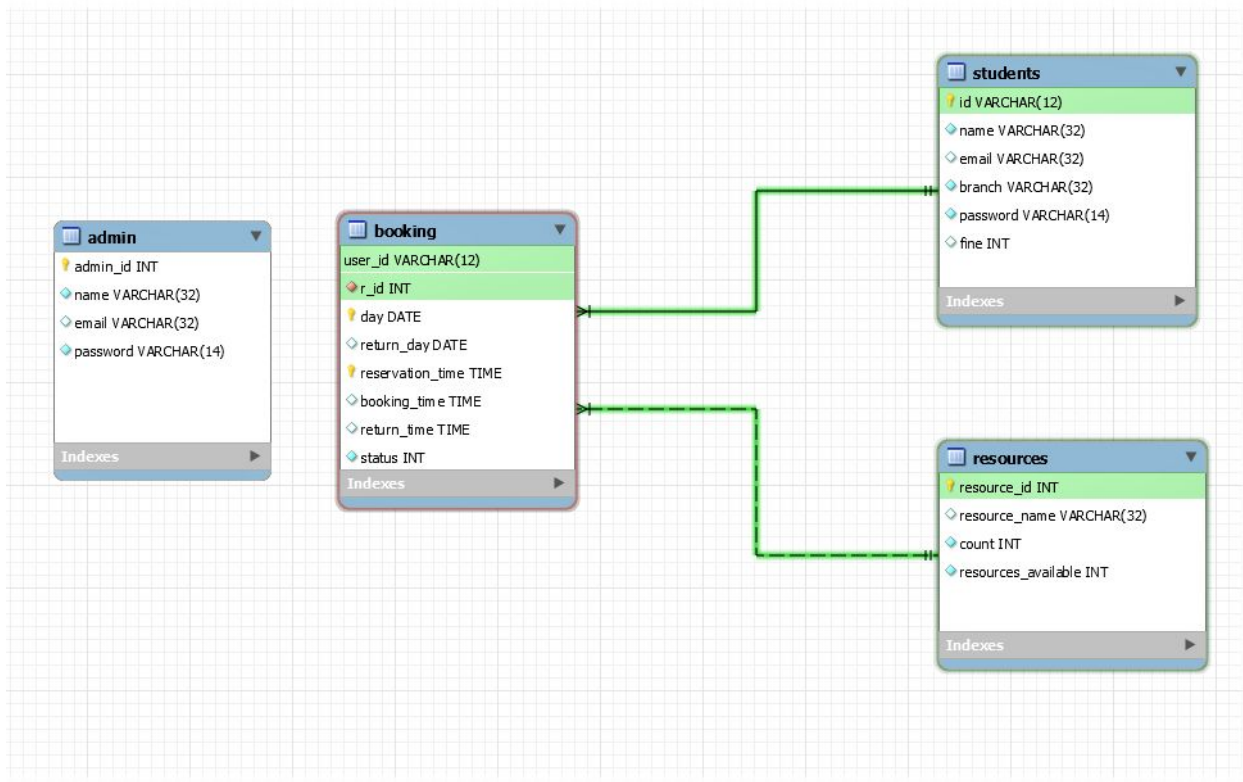
- Lombik modul

3.4.3 Hardverkövetelmény

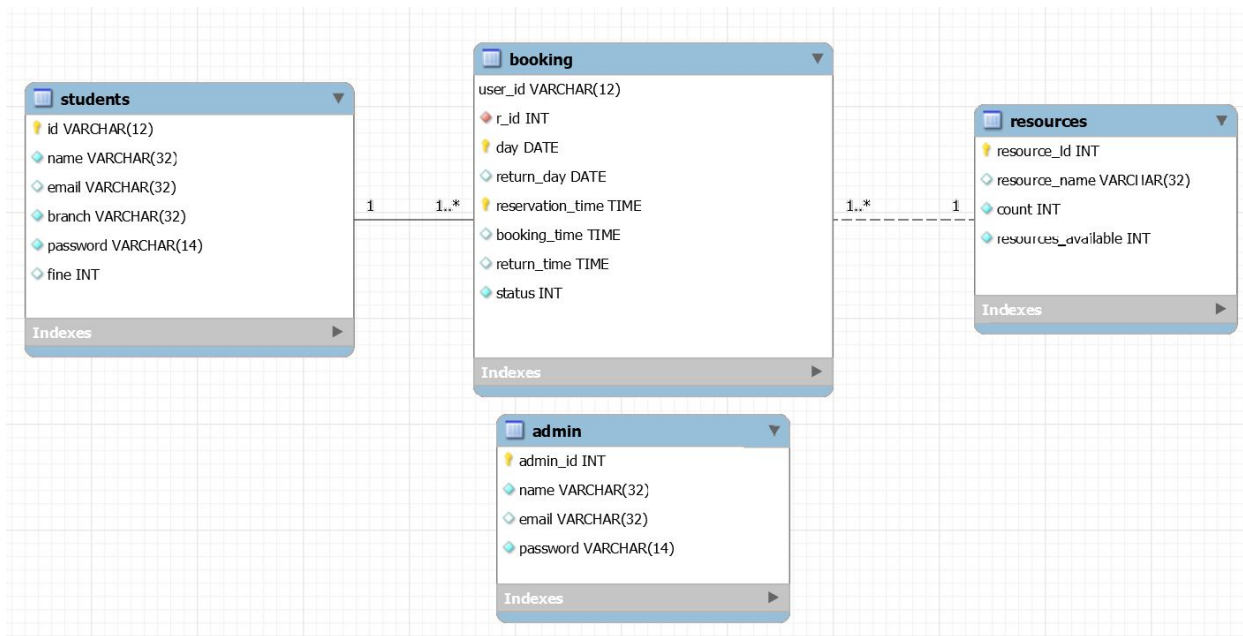
-)Modern operációs rendszer:
 - Windows 7 vagy 10
 - Mac OS X 10.11 vagy újabb, 64 bites
 - Linux: RHEL 6/7, 64 bites (majdnem minden könyvtár működik Ubuntuban is)
- x86 64 bites CPU (Intel / AMD architektúra)
- Minimum 4 GB RAM
- Minimum 5 GB szabad lemezterület
- Az Android 2.3xGingerbread és újabb verzióihoz 64 bites környezet szükséges.
- Legalább 250 GB szabad lemezterület a kód megtekintéséhez és további 150 GB a megépítéséhez.
- Legalább 8 GB RAM/csere

4. A JAVASOLT RENDSZER TERVEZÉSE

4.1 Entitáskapcsolat diagram

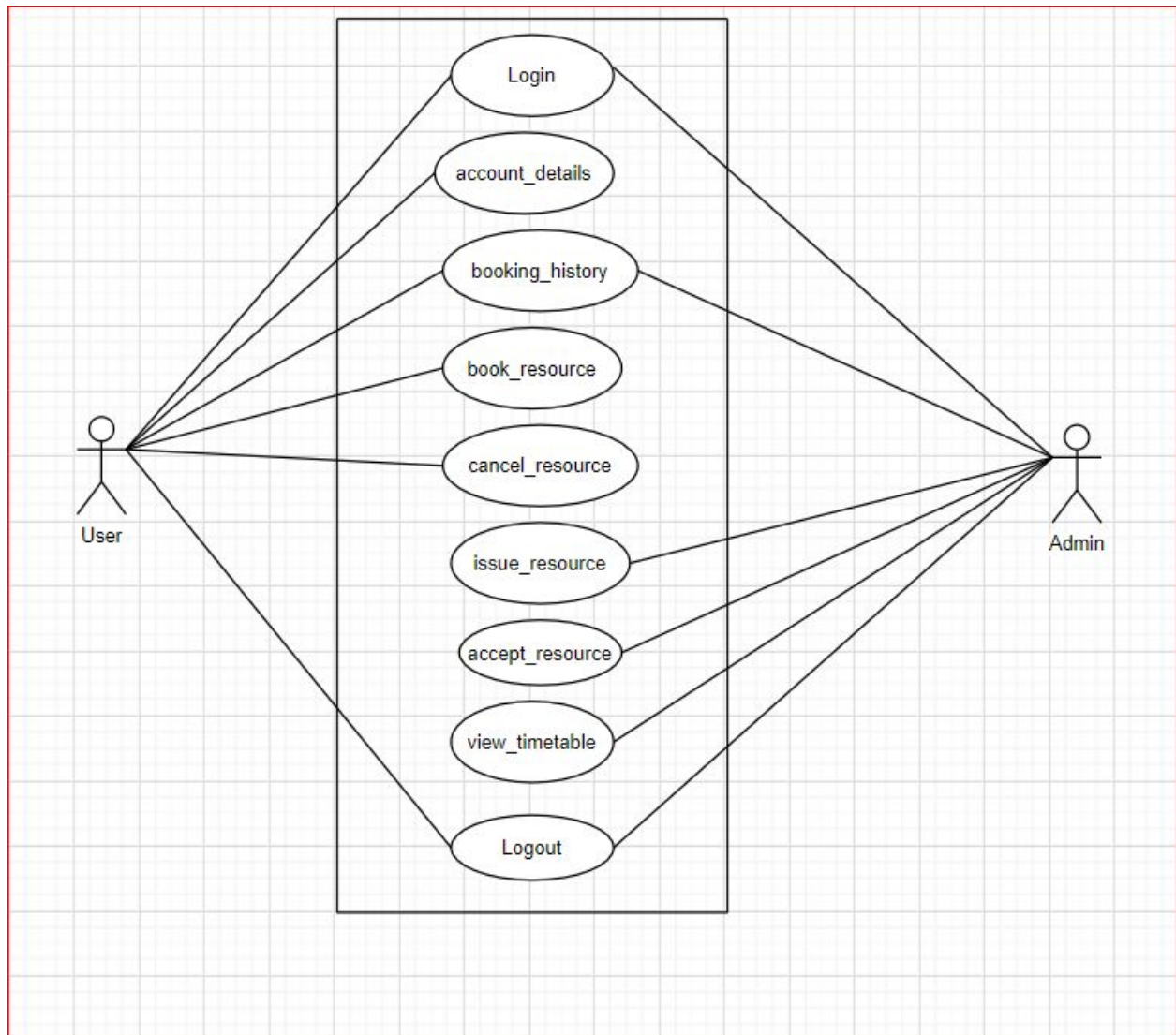


1. ábra: Entitáskapcsolati diagram



4.2 UML diagramok

4.2.1 Használati eset diagram



2. ábra: Használati eset diagram

4.3 Adatbázis tervezés

Diákok asztala

	id	name	email	branch	password	fine
▶	160118733012	Prathyusha	sakuraprunus1@gmail.com	cse	abc123	0
	160119733082	Amith	amith2610@gmail.com	cse	xyz123	50
	160119733103	Koppula Sai Charan	koppulasaicharan001@gmail.com	cse	abc123	0
•	NULL	NULL	NULL	NULL	NULL	NULL

3. ábra: Diákok táblázata

Foglalási asztal

user_id	r_id	day	return_day	reservation_time	booking_time	return_time	status
160117733001	103	2020-06-30	NULL	14:39:53	14:50:58	NULL	1
160117733115	103	2020-06-23	NULL	12:15:00	NULL	NULL	0
160117733115	100	2020-06-23	NULL	13:00:00	NULL	NULL	0
160117733115	103	2020-06-24	2020-06-29	12:15:00	12:25:00	11:13:16	1
160117733115	103	2020-06-25	NULL	12:15:00	12:25:00	NULL	2
160117733116	103	2020-06-22	2020-06-24	12:15:00	12:20:00	15:44:38	1
160117733116	103	2020-06-24	2020-06-29	12:15:00	12:25:00	11:13:18	1
160118733011	103	2020-06-22	NULL	14:15:00	NULL	NULL	2
160118733011	102	2020-06-30	NULL	14:41:44	NULL	NULL	2
160118733012	100	2020-06-20	2020-06-20	14:20:00	14:30:00	15:20:00	1
160118733012	101	2020-06-21	2020-06-23	12:20:00	NULL	16:41:05	1
160118733012	100	2020-06-22	2020-06-22	12:10:00	12:20:00	13:20:00	1
160119733082	104	2020-06-26	2020-06-30	12:10:00	NULL	14:25:39	1
160119733082	101	2020-06-27	NULL	12:10:00	NULL	NULL	2
160119733082	101	2020-06-28	NULL	12:56:58	NULL	NULL	1

4. ábra: Foglalási táblázat

Erőforrások táblázata

	resource_id	resource_name	count	resources_available
▶	100	Basket Balls	27	25
	101	Cricket Bats	29	28
	102	Cricket balls	30	30
	103	Footballs	40	39
	104	Badminton Rackets	33	31
	105	Chess	30	30
	106	Carrom coins	30	30
*	NULL	NULL	NULL	NULL

5. ábra: Erőforrások táblázata

Admin táblázat

	admin_id	name	email	password
▶	100	Admin	NULL	abc123
	101	Admin	sakuraprunus1@qm...	abc123
*	NULL	NULL	NULL	NULL

6. ábra: Adminisztrációs táblázat

4.4 Modul leírása

Ebben a projektben két modul található. Ezek a felhasználói és az adminisztrátorok.

4.4.1 A felhasználó jellemzői

A felhasználó egy diák, aki a „SportEasy” Android alkalmazást fogja használni. Az alkalmazás a következő funkciókkal rendelkezik:

- Megtekintheti a sportblokkban rendelkezésre álló összes erőforrást, annak számával az alkalmazásban. Ez csökkenti a kérdésszámlálónál a lekérdezési időt.
- Egyetlen kattintással elküldheti a foglalási kérelmet, így 20 percen belül átveheti foglalási ideje.
- A felhasználónak lehetősége van a foglalást követő 20 percen belül lemondani a foglalási kérelmét idő.

- kéréseket fogalmazott meg.

A felhasználónak lehetősége van új jelszót beállítani, ha elfelejti jelszavát vagy meg akarja változtatni.

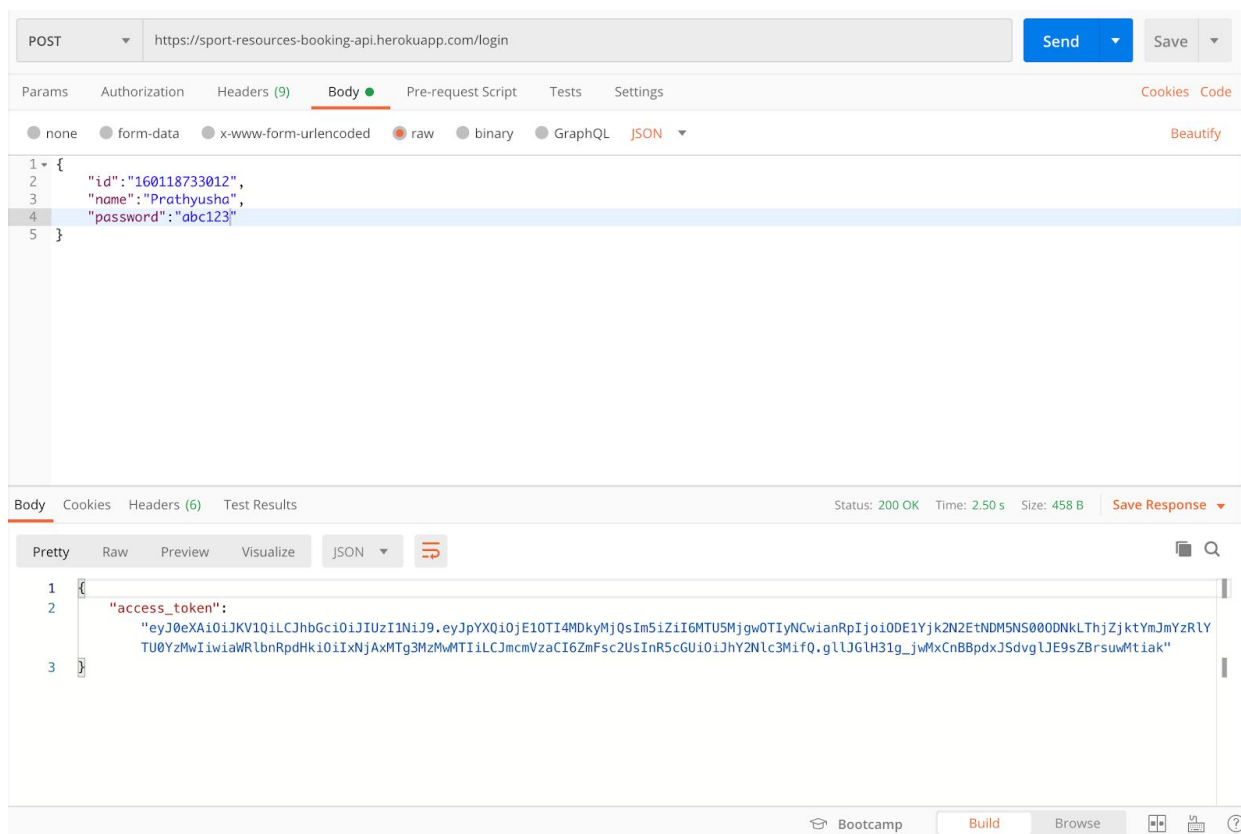
4.4.2 Az API jellemzői

Ennek a Flask RESTful API for Sports erőforrás-foglalási alkalmazásnak a fő célja, hogy a diákok lehessen foglalni a sportforrásokat, az adminisztrátorok pedig nyomon követhessék az erőforrások elérhetőségét, és egyszerűsítsék a finom folyamatot.

Link:<https://sport-resources-booking-api.herokuapp.com/>

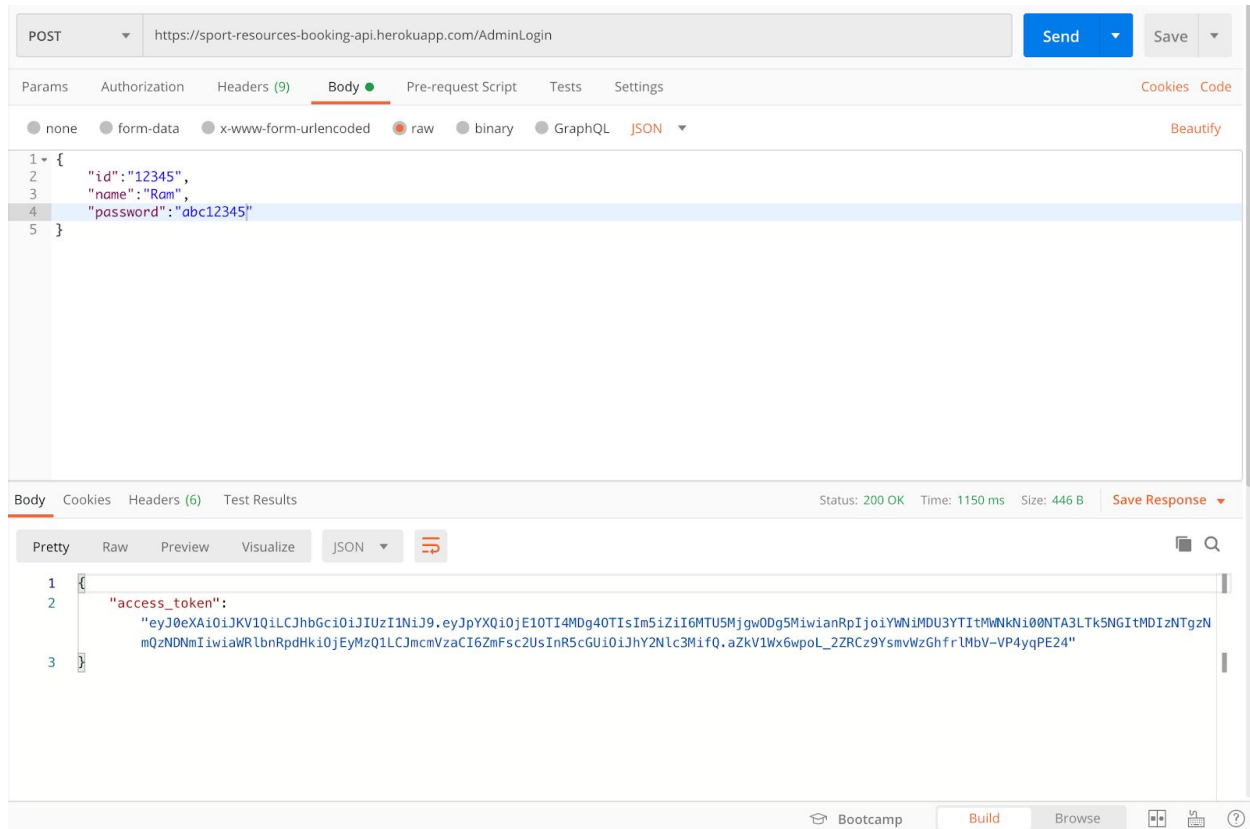
Az adminisztrátor és a felhasználó a /adminLogin,/login végpontokat használhatja a bejelentkezéshez:

a) /login POST Request – egy JSON objektumot vesz fel 'id', 'username' és 'password' értékkel, és visszaadja a JWT token, ha létezik a Users táblában. A JWT tokenet kell használni az összes végpont eléréséhez. Az összes végponthoz fel kell tüntetni egy engedélyezési fejléctet, amelynek értéke „hordozó”.



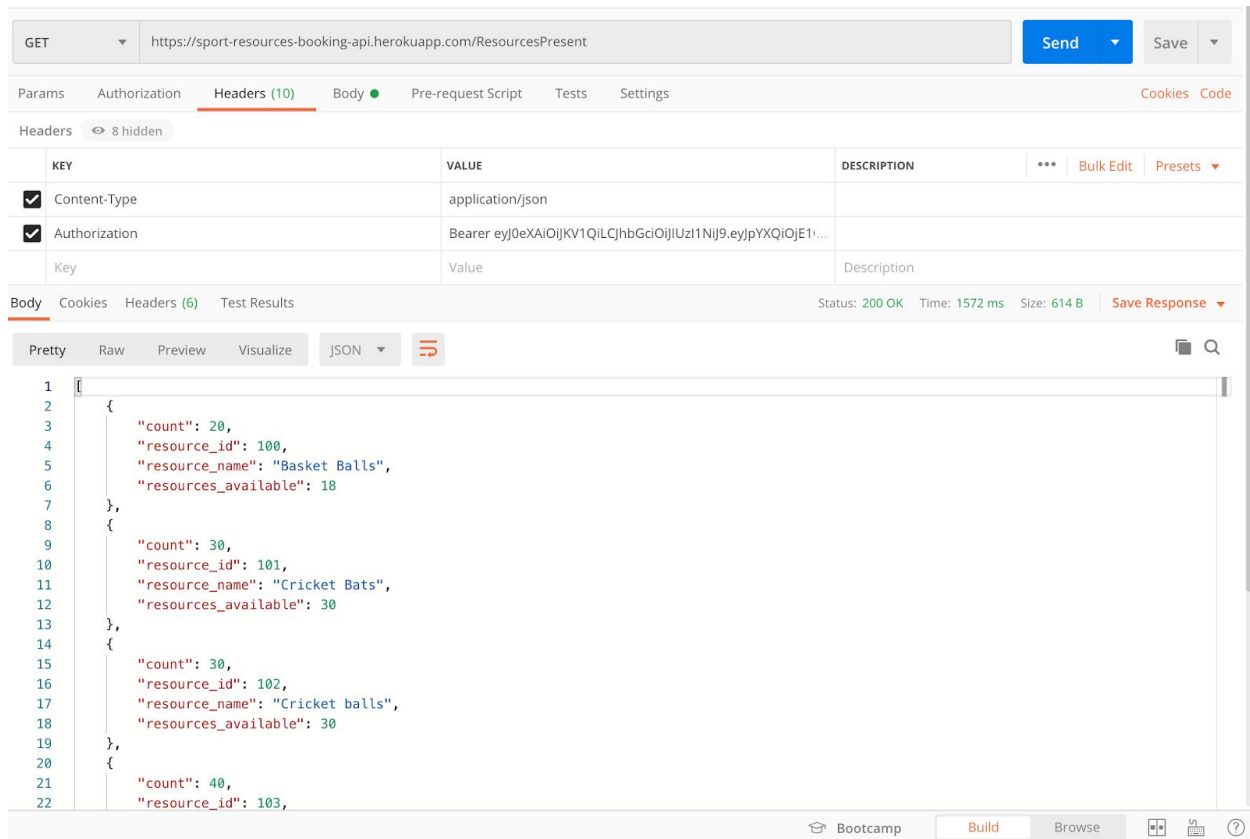
7. ábra: Bejelentkezési végpont

b) /adminLogin - POST Request -végpont csak az adminoknak szól, és pillanatnyilag egy adminisztrátor van az adminisztrációs táblában, aki hozzáférhet a webalkalmazáshoz. Ezenkívül egy „felhasználónévvel” és „jelszóval” rendelkező JSON-objektumot vesz fel, és visszaad egy JWT-jogkivonatot, ha létezik az Admin táblázatban. A JWTtoken-t kell használni az összes végpont eléréséhez. Az összes végponthoz fel kell tüntetni egy engedélyezési fejléct, amelynek értéke „hordozó”.



8. ábra: Rendszergazdai bejelentkezési végpont

/ResourcesPresent- GET request- az erőforrástáblából, hogy részleteket adjon meg az összes jelenlévő erőforrásról.

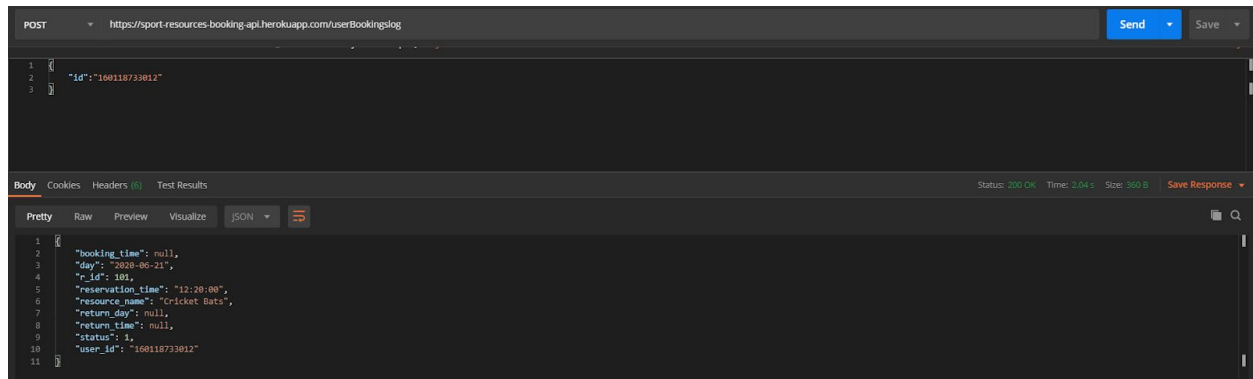


9. ábra: Erőforrások végpontja

`/AddExtraResource`- Az adminisztrátor csak a hozzáférési token megadása után hajthat végre módosításokat. Frissíti az erőforrástáblát, és beszúr egy erőforrást, amikor az adminisztráció új erőforrást vásárol.

`/DeleteResource`- Az adminisztrátor csak a hozzáférési token megadása után hajthat végre módosításokat. Frissíti az erőforrástáblát és törli az id által megadott erőforrást.

`/userBookingslog` – A felhasználó által még nem visszaküldött vagy a felhasználó által lefoglalt foglalás részleteit tartalmazza



10. ábra: Felhasználói foglalási napló

/userDue-GET kérés egy hallgató esedékességének megszerzéséhez, ha van ilyen, és az erőforrás adatai, amelyet nem küldtek vissza

/cancelBooking- POST kérés, hogy először ellenőrizze, hogy a foglalás létezik-e, majd módosítsa az állapotát lefoglalt/elutasítottra, és üzenetet küld a felhasználónak.

/resourceDetails- GET kérés, hogy egy adott erőforrás összes részletét megkapja az erőforrástáblából egy azonosító segítségével.

/incrementByOne-Az adminisztrátor csak a hozzáférési token megadása után hajthat végre módosításokat. Frissíti az erőforrástáblázatot, és 1-gyel növeli a számértéket, ha új erőforrást vásárol az adminisztráció.

/incrementByValue- Az adminisztrátor csak a hozzáférési token megadása után hajthat végre módosításokat. Frissíti az erőforrás-táblázatot, és növeli a számlálási értéket, amikor az adminisztráció új erőforrást vásárol.

/decrementByOne- Az adminisztrátor csak a hozzáférési token megadása után hajthat végre változtatásokat. Frissíti az erőforrástáblázatot, és 1-gyel csökkenti a számértéket, amikor az adminisztráció új erőforrást vásárol.

/decrementByValue- GET kérés, amelyet az adminisztrátor csak a hozzáférési jogkivonat megadása után hajthat végre. Frissíti az erőforrástáblát, és csökkenti a számértéket, amikor az adminisztráció új erőforrást vásárol.

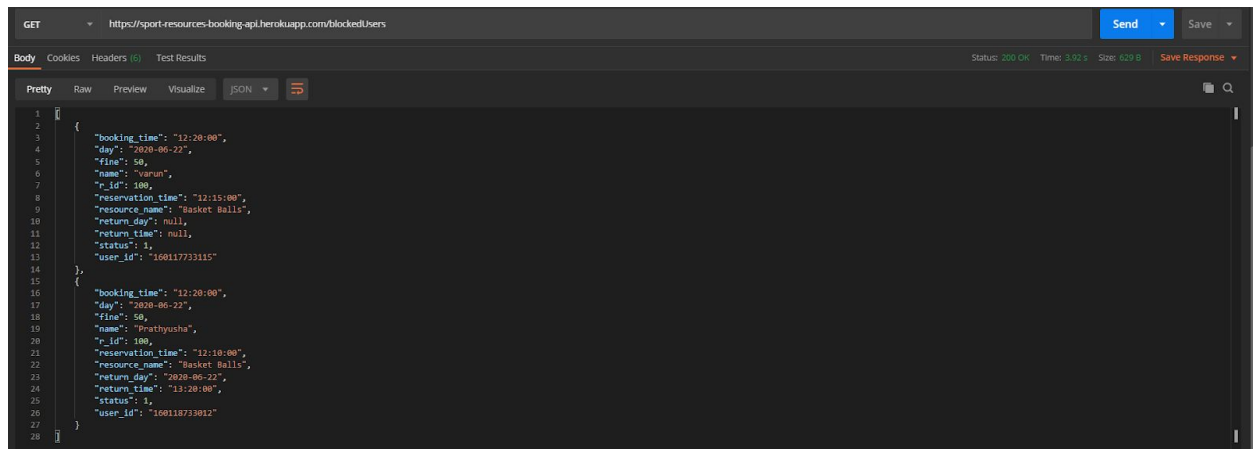
Az /issueResource- frissíti a booking_time értéket, és az állapotot egyre állítja

Az /acceptResource- frissíti a rendelkezésre álló erőforrásokat, és automatikusan beállítja a bírságot, ha nem tér vissza 16:20 után.

/bookingHistory-GET kérés a BookingHistory1 nézetből végrehajtott összes foglalás naplójának lekéréséhez.

/issuedBookings-GET kérés az aktuális napon kiadott összes foglalás naplójának lekérésére a BookingHistory1 nézetből.

/blockedUsers- Kérés a letiltott felhasználók fogadására, hogy ne adjanak ki több erőforrást



11. ábra: Letiltott felhasználók

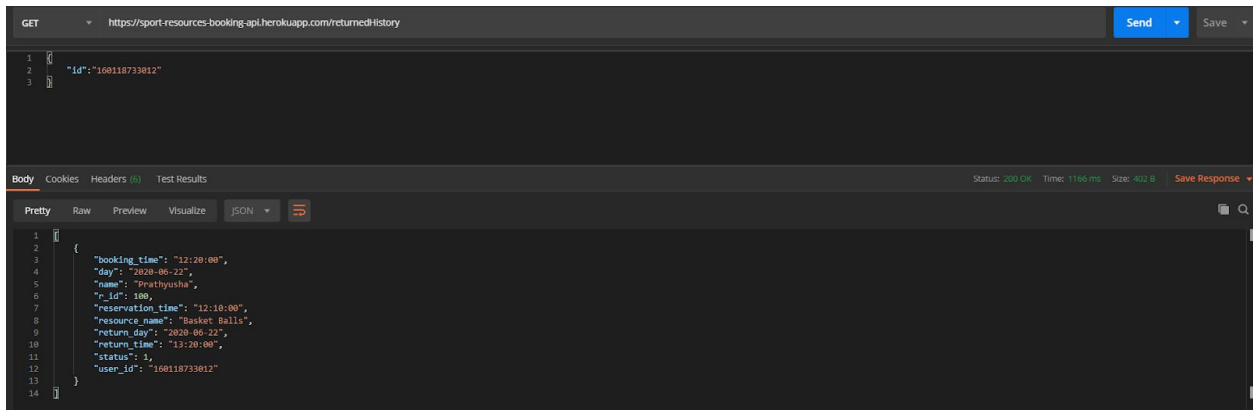
/unlockUser-GET kéri az adatbázis frissítését a felhasználó státuszának lekéréséhez és a felhasználó esedékességi állapotának megváltoztatásához, amikor a felhasználó fizeti díját.

/blockUser-GET kéri a bírság frissítését az azonosított használok felhasználó azonosítója mellett.

A /bookResource egy erőforrás lefoglalására szolgál a felhasználó által, így a részletek hozzáadódnak az adatbázishoz

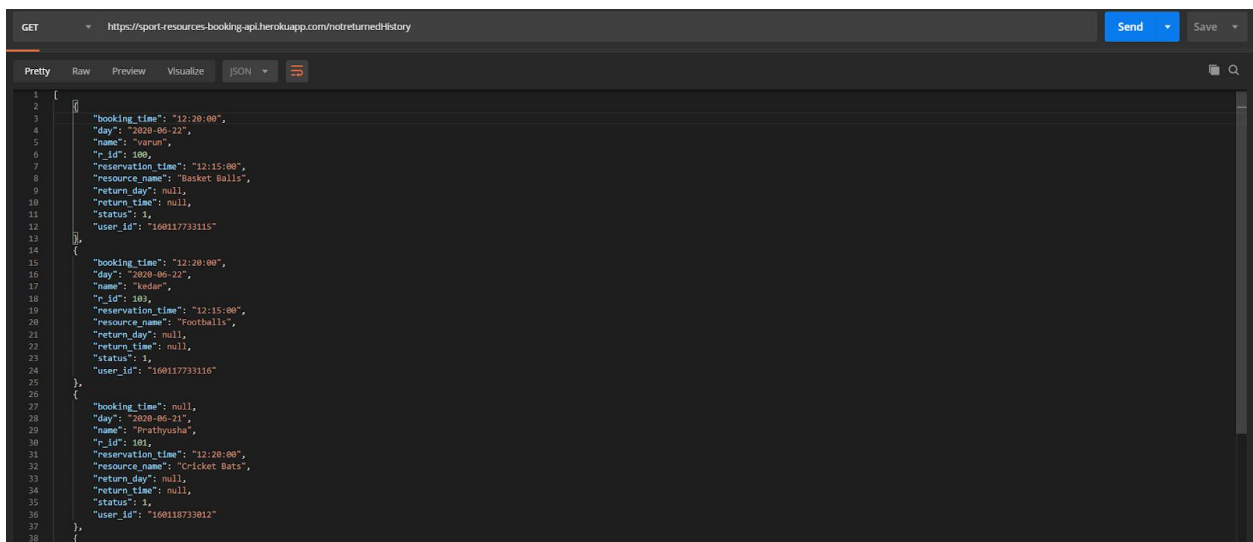
/rejectBooking-GET kérés egy felhasználó foglalásának elutasítására, ha az adminisztrátor akarja, az órarenddel kapcsolatos problémák miatt.

/returnedHistory-GET kérés az összes olyan felhasználó naplójának lekéréséhez, aki a BookingHistory2 nézetből visszaküldte a mai napra lefoglalt erőforrásokat.



12. ábra: Visszaadott előzmények

/notreturnedHistory- KÉRJÜK le kérés az összes olyan felhasználó naplójának lekéréséhez, aki addig nem küldte vissza a lefoglalt erőforrásokat a BookingHistory2 nézetből.



13. ábra: Nem visszaadott előzmények

/forgot_password -GET kérés Ha a felhasználó jelszómódosítási kérelmet küld az azonosítóval együtt Ez megerősítő e-mailt küld a jelszó megváltoztatásához az ajánlott levél azonosítójára.

/update_password -GET kérés, a felhasználó megváltoztathatja jelszavát az id, régi_password és new_password.

4.4.3 Az adminisztrátor jellemzői

Az adminisztrátor az, aki elfogadja és jóváhagyja a foglalási kérelmeket. Ők kezelik a webet alkalmazás. Az adminisztrációs modul funkciói a következők:

- Szükség esetén az erőforrás hozzáadása és törlése.
- A felhasználó által igényelt erőforrás elfogadása vagy elutasítása az órarend alapján.
- A foglalás frissítése attól függően, hogy a forrásokat visszaküldték-e vagy sem, és ennek megfelelően bírságokat szabnak ki.
- A felhasználó blokkolása, ha az erőforrás sérült.
- A bírság kézi beszedése és a felhasználó blokkolásának feloldása.

5. MEGVALÓSÍTÁS ÉS EREDMÉNY

5.1 pszeudo kód:

Ez a django kód a webalkalmazásban található összes erőforrás megjelenítéséhez.

```
def api_call (kérésére):
    )globális
    )ha(p):
        adat =
requests.get( "https://sport-resources-booking-api.herokuapp.com/ResourcesP resent" ,fejlécek= {
'Engedélyezés':f'hordozó{ o} '}))
        res = data.json()
        )ha(p== "):
            )visszaredirect ('bejelentkezés') )
        mást:
            kontextus={ 'adatok': res,}
            )visszarend(request, 'api.html',kontextus)
        )mást:
            )visszaredirect ('bejelentkezés')
```

Ez a django api híváskódja a foglalási kérelmek oldalához:

```
def foglalási_kérések(kérésére):
    )globális
    )ha(p):
        keresési_kifejezés ="
        )ha)'keresés')bekérés.GET:
            keresési_kifejezés = kérés.GET[ 'keresés']
        td =
requests.get( "https://sport-resources-booking-api.herokuapp.com/bookingReq uests" ,fejlécek= {
'Engedélyezés':f'hordozó{ o} },
        )adatok={ 'keresés'
:keresési_kifejezés}) res = td.json()
        kontextus={ 'adatok': res,}
        )visszarend(request, 'bookingreq.html' ,kontextus) )mást:
            )visszaredirect ('bejelentkezés')
```

```

42
43
44
45 api.add_resource(UserLogin, '/login')
46 api.add_resource(AdminLogin, '/AdminLogin')
47 api.add_resource(ResourcesPresent, '/ResourcesPresent')
48 api.add_resource(User_Bookings_log, '/userBookingslog')
49 api.add_resource(UserBookingFine, '/userDue')
50 api.add_resource(cancelBooking, '/cancelBooking')
51 api.add_resource(resourceDetails, '/resourceDetails')
52 api.add_resource(incrementResourcesByOne, '/incrementByOne')
53 api.add_resource(incrementResourcesByValue, '/incrementByValue')
54 api.add_resource(decrementResourcesByOne, '/decrementByOne')
55 api.add_resource(decrementResourcesByValue, '/decrementByValue')
56 api.add_resource(issueResource, '/issueResource')
57 api.add_resource(acceptReturnedResource, '/acceptResource')
58 api.add_resource(bookingHistory, '/bookingHistory')
59 api.add_resource(issuedBookings, '/issuedBookings')
60 api.add_resource(blockedUsers, '/blockedUsers')
61 api.add_resource(unblockUser, '/unblockUser')
62 api.add_resource(blockUser, '/blockUser')
63 api.add_resource(bookResource, '/bookResource')
64 api.add_resource(bookingRequests, '/bookingRequests')
65 api.add_resource(rejectBooking, '/rejectBooking')
66 api.add_resource(returnedHistory, '/returnedHistory')
67 api.add_resource(notreturnedHistory, '/notreturnedHistory')
68 api.add_resource(notreturnedToday, '/notreturnedToday')
69 api.add_resource(allBookings, '/allBookings')
70 api.add_resource(AddExtraResource, '/AddExtraResource')
71 api.add_resource(DeleteResource, '/DeleteResource')
72 api.add_resource(Users, '/users')
73 api.add_resource(changePassword, '/changePassword')
74 api.add_resource(timetable, '/timetable')
75 api.add_resource(admin_change_password, '/admin_change_password')

```

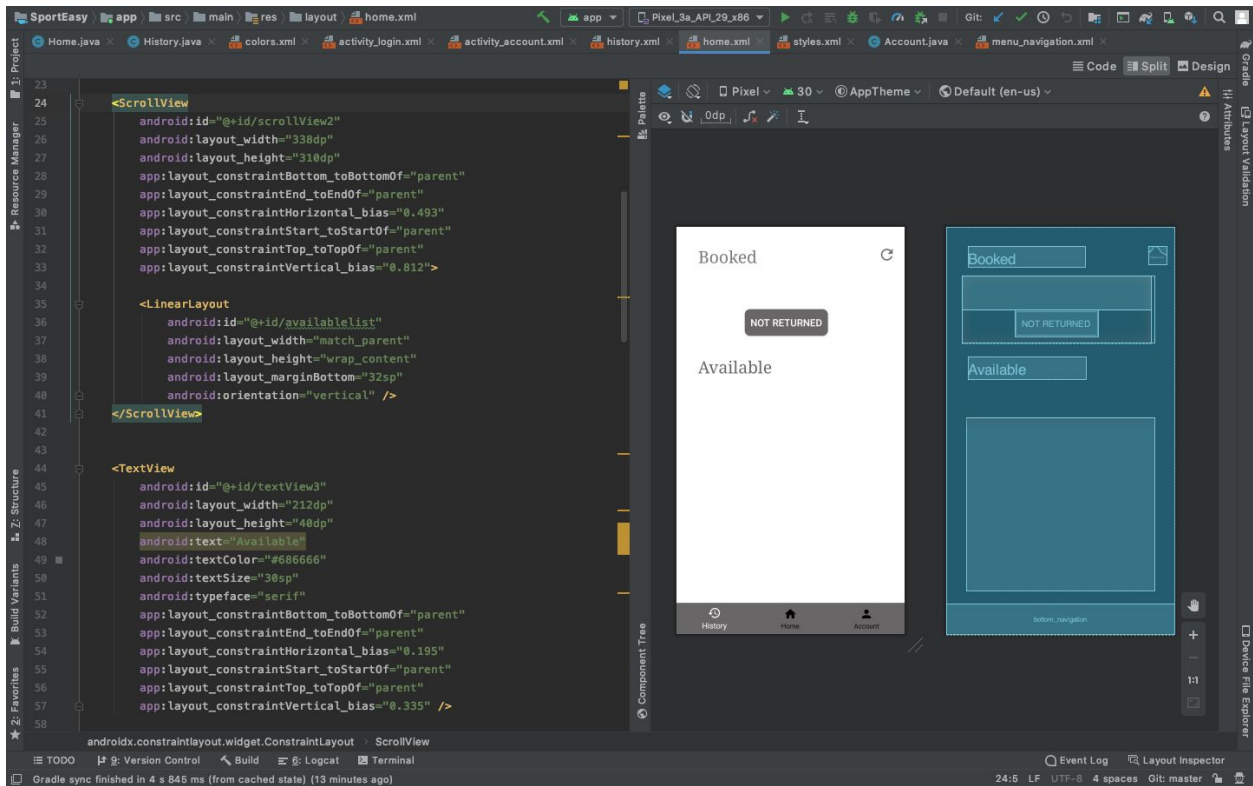
```

class UserLogin(Resource):
    parser=reqparse.RequestParser()
    parser.add_argument('id',type=str,required=True,help="ID cannot be blank.")
    parser.add_argument('password',type=str,required=True,help="Password cannot be blank.")
    def post(self):
        data=self.parser.parse_args()
        user=User.getUserById(data['id'])
        if user and safe_str_cmp(user.password,data['password']):
            access_token=create_access_token(identity=user.id,expires_delta=False)
            return {'access_token':access_token},200
        return {"message":"Invalid Credentials!"}, 401

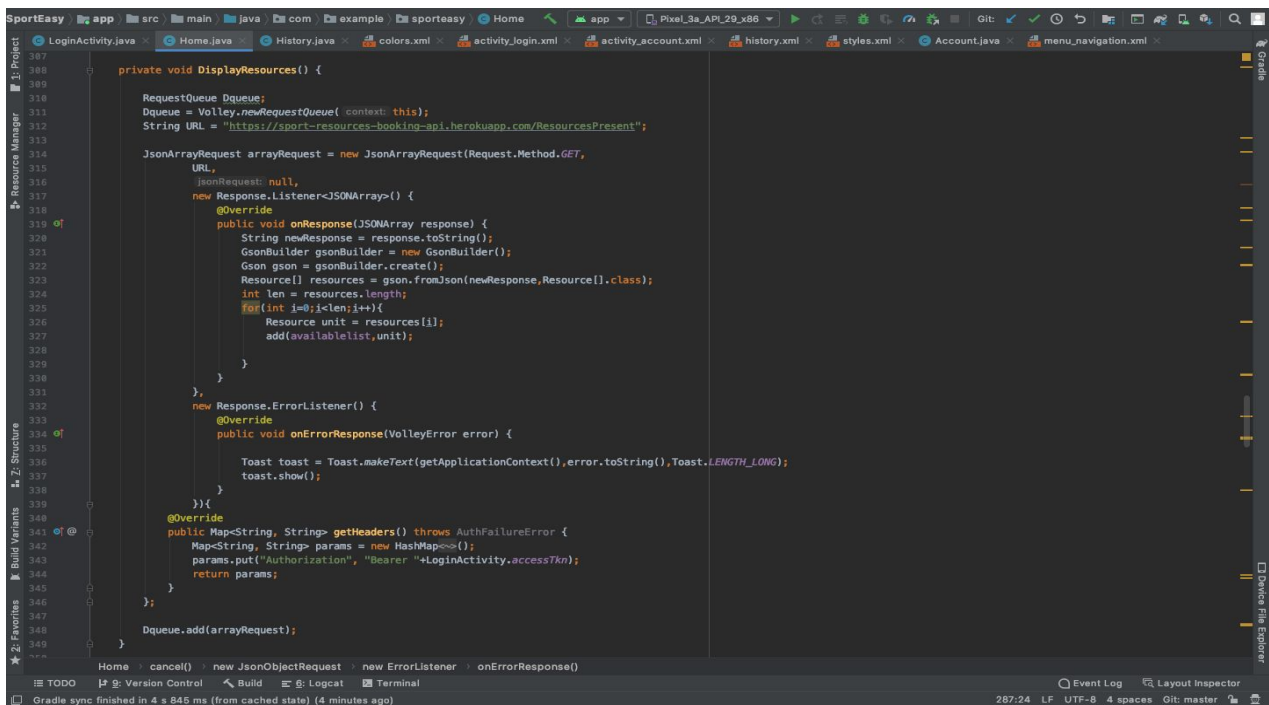
class changePassword(Resource):
    @jwt_required
    def post(self):
        parser=reqparse.RequestParser()
        parser.add_argument('id', type=str, required=True, help='user_id Cannot be blank')
        parser.add_argument('password',type=str,required=True,help="Password cannot be blank.")
        parser.add_argument('new_password',type=str,required=True,help="new Password cannot be blank.")
        data= parser.parse_args()
        try:
            res = query(f"""Select * from students where id='{data["id"]}' """,return_json=False)
            if(res[0]['password']==data['password']):
                if(res[0]['password']!=data['new_password']):
                    query(f""" update students set password='{data['new_password']}' where id='{data['id']}' """)
                    return {"message":"password changed !"},200
                else:
                    return {"message":"old password and new are same,cannot update !"},200
            else:
                return {"message":"enter the correct password"},401
        except:
            return {"message": "There was an error connecting to user table"}, 400

```

Xml-kód a megjelenített erőforrások elrendezéséhez.



Java kód az elérhető sportforrások megjelenítéséhez.



5.2 Tervezési lépések és kimenetek/eredmények

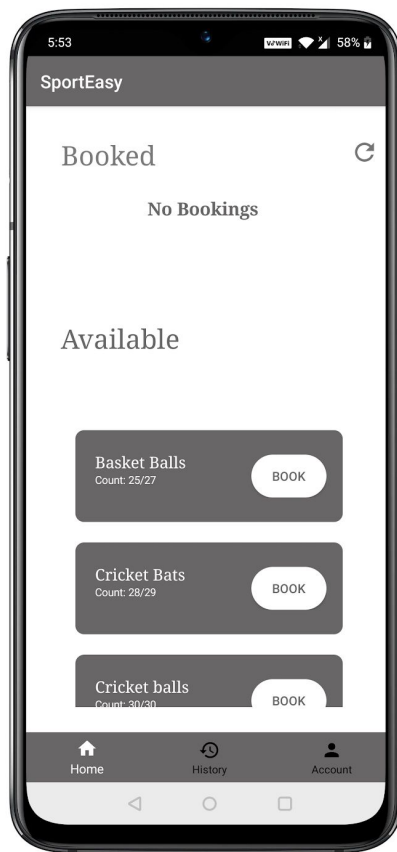
5.2.1 Mobil alkalmazás:

5.2.1.1 Bejelentkezési tevékenység:A hallgató a névjegyzékszámával és jelszavával jelentkezhet be ebbe az alkalmazásba



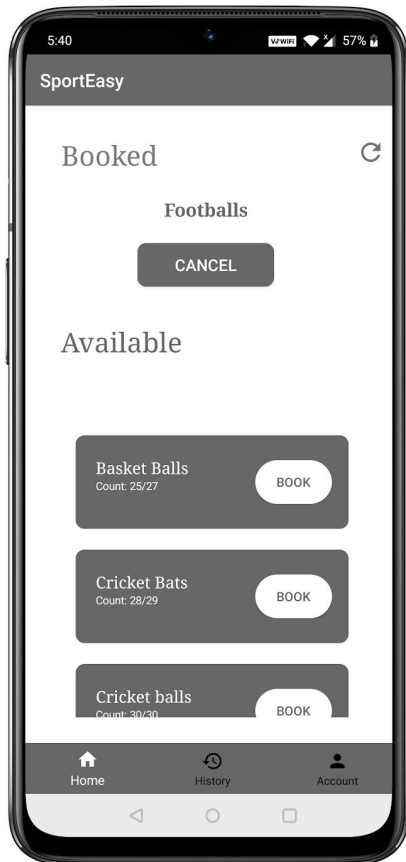
14. ábra: Felhasználó bejelentkezési tevékenysége

5.2.1.2 Otthoni tevékenység: Ez a sportblokkban és a sportblokkban található erőforrások részleteit mutatja a felhasználó aktuális foglalásának adatait is

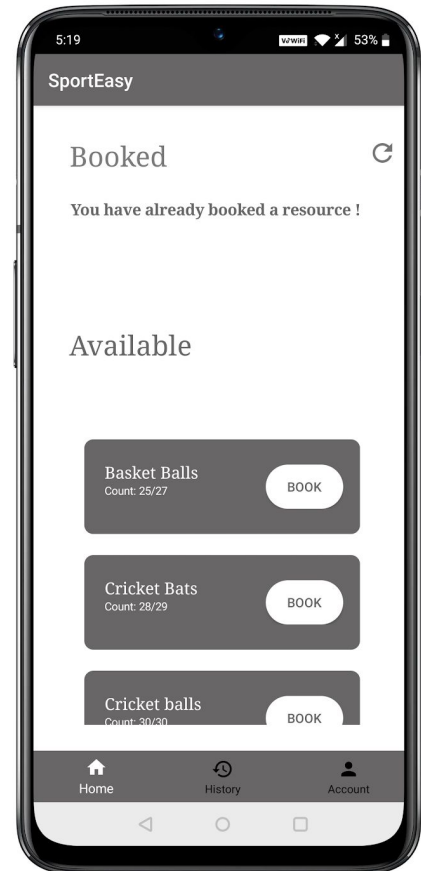


15. ábra: Felhasználói kezdőlap 1

5.2.1.3 Foglalási erőforrás oldal a felhasználó számára:A felhasználó 23:40 között foglalhat le erőforrást és 15:00, és ha nincs pénzbírsága vagy forrása, amelyet még nem küldtek vissza. És lemondhatja 20 percen belül

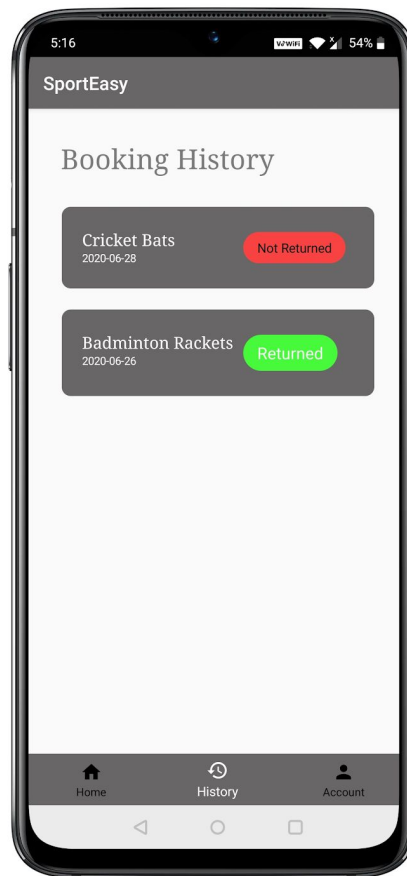


16. ábra: Felhasználói kezdőlap 2



17. ábra: Felhasználói kezdőlap 3

5.2.1.4 Foglalási előzmények oldal a felhasználó számára:Felsorolja a felhasználó által végrehajtott összes foglalást vissza, nem hozták vissza, nem gyűjtötték.



18. ábra: A felhasználó foglalási előzményei

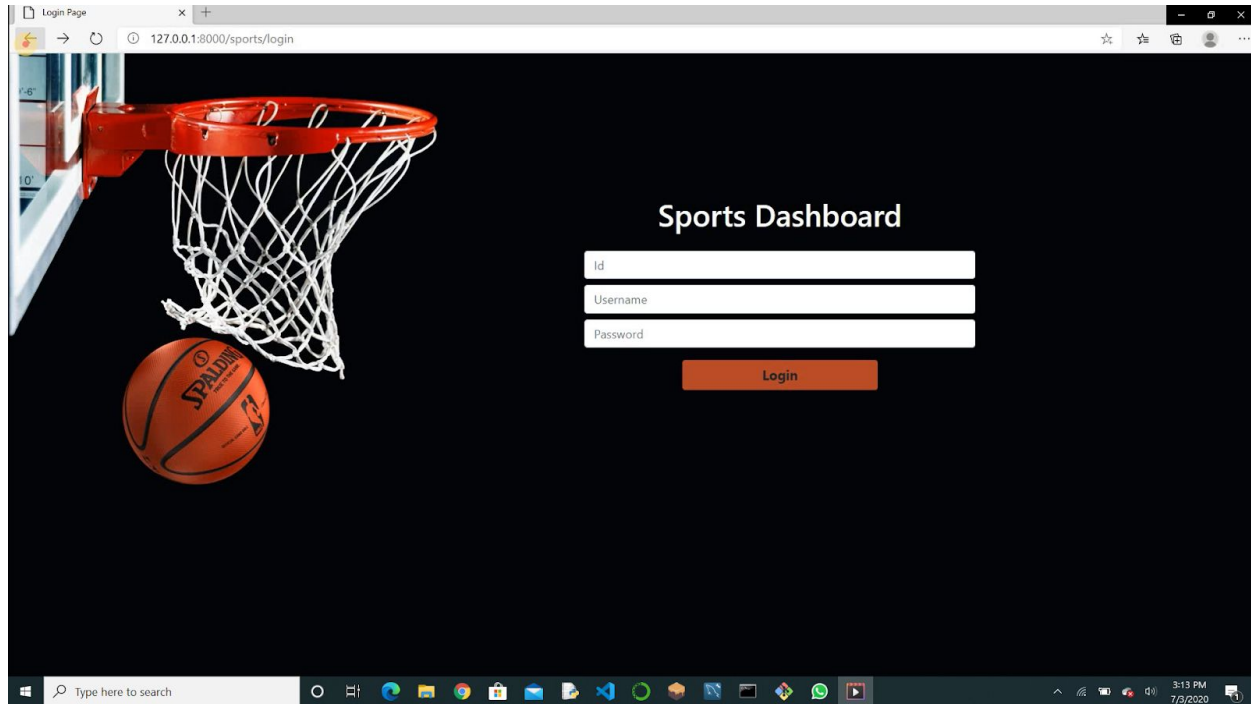
5.2.1.5 A felhasználó fiókoldala: Ez a felhasználó adatait jeleníti meg, és lehetősége van rá jelszó módosítása és kijelentkezés.



19. ábra: Felhasználó fiókja

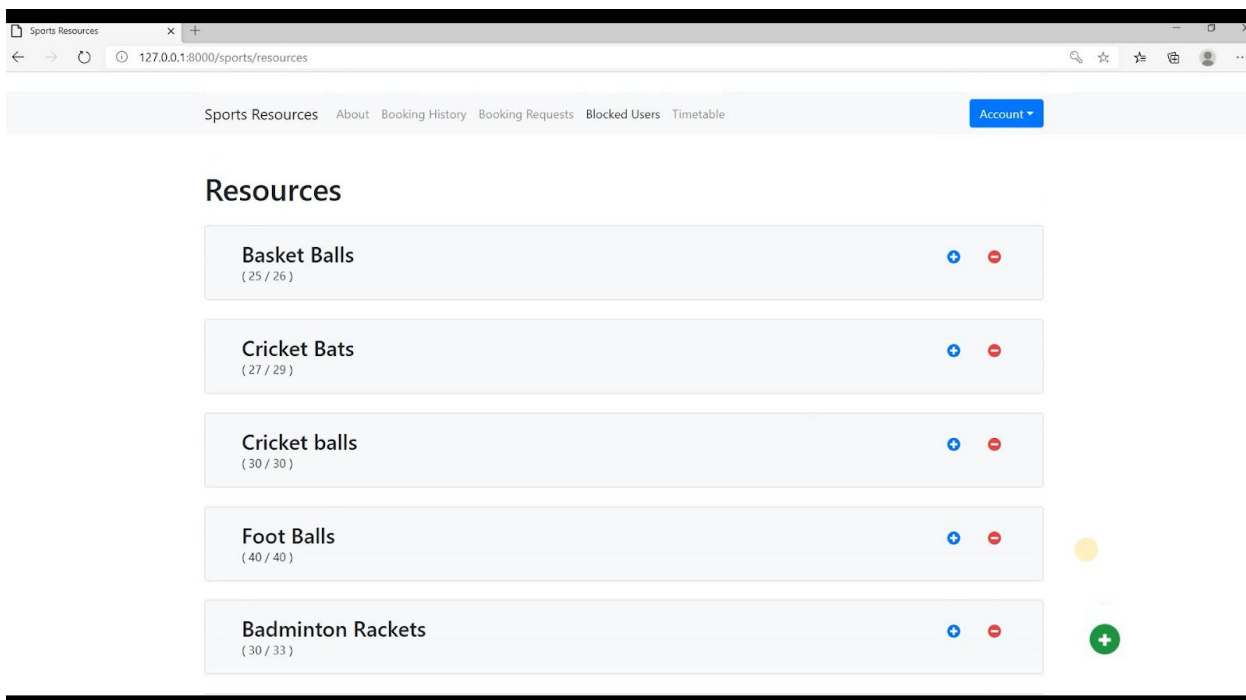
5.2.2 Webes alkalmazás:

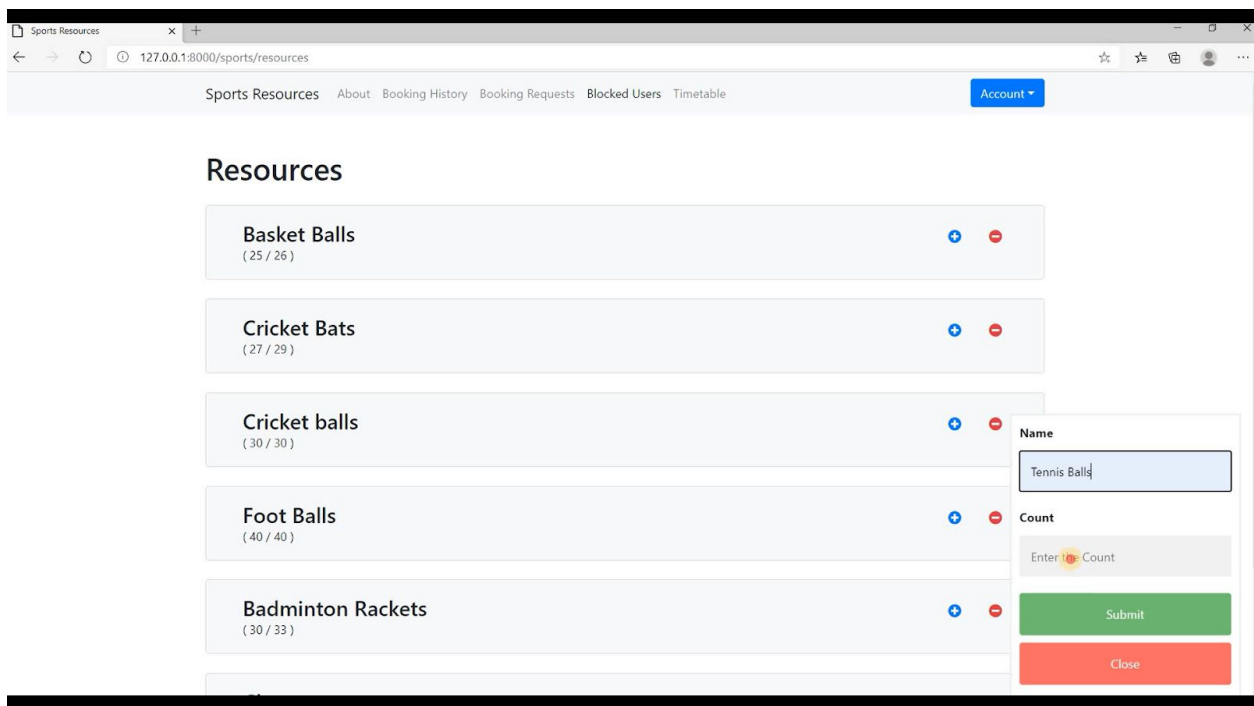
5.2.2.1 Bejelentkezési oldal:A bejelentkezési oldal lehetővé teszi az adminisztrátor számára, hogy bejelentkezzen a webalkalmazásba a megfelelő adatok megadásával a megadott mezőkbe.



20. ábra: Bejelentkezési oldal

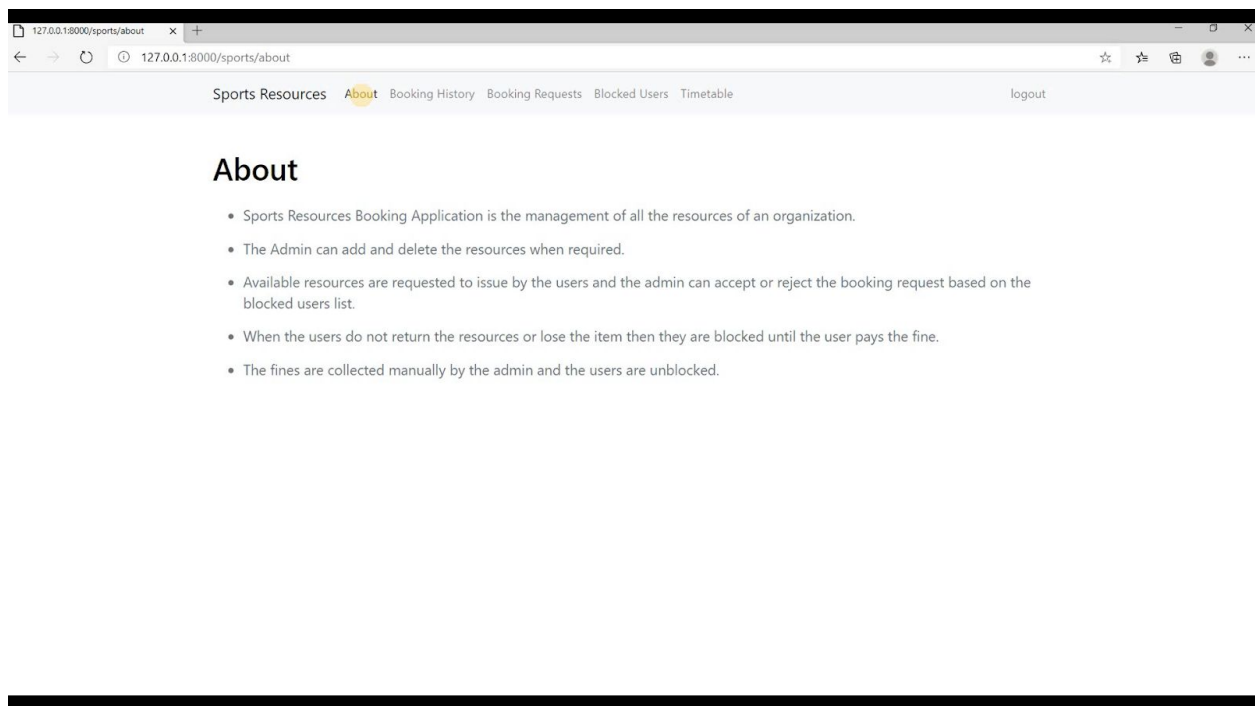
5.2.2.2 Kezdőlap (Sportforrások oldal):Ez az oldal az elérhető erőforrások listáját jeleníti meg. Az erőforrások és a számuk a plusz és mínusz gombokkal frissíthető.





21. ábra: Kezdőlap (Források oldala)

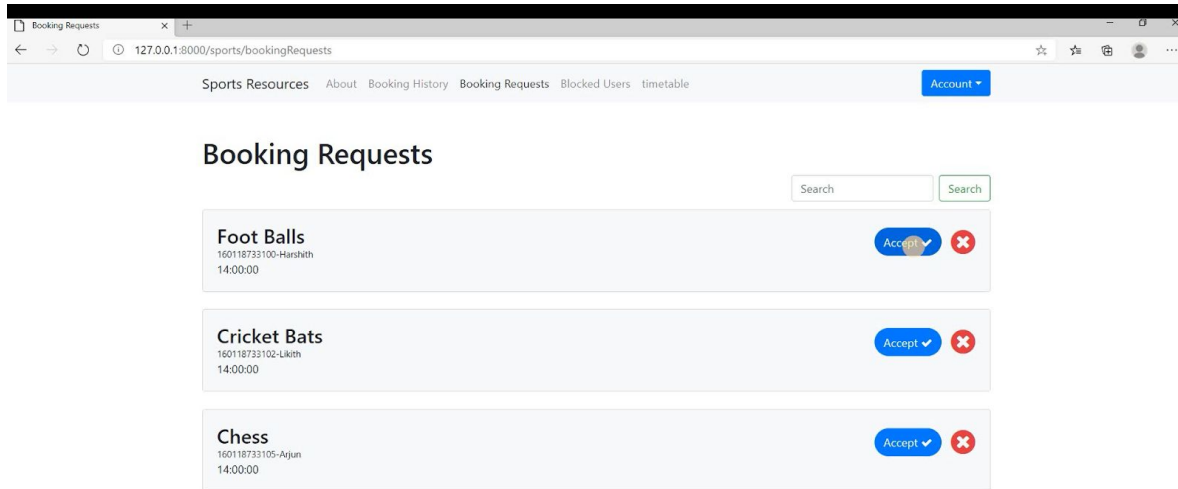
5.2.2.4 Az oldal névjegye: Ez az oldal az alkalmazással kapcsolatos információkat jelenít meg.



22. ábra: Az oldalról

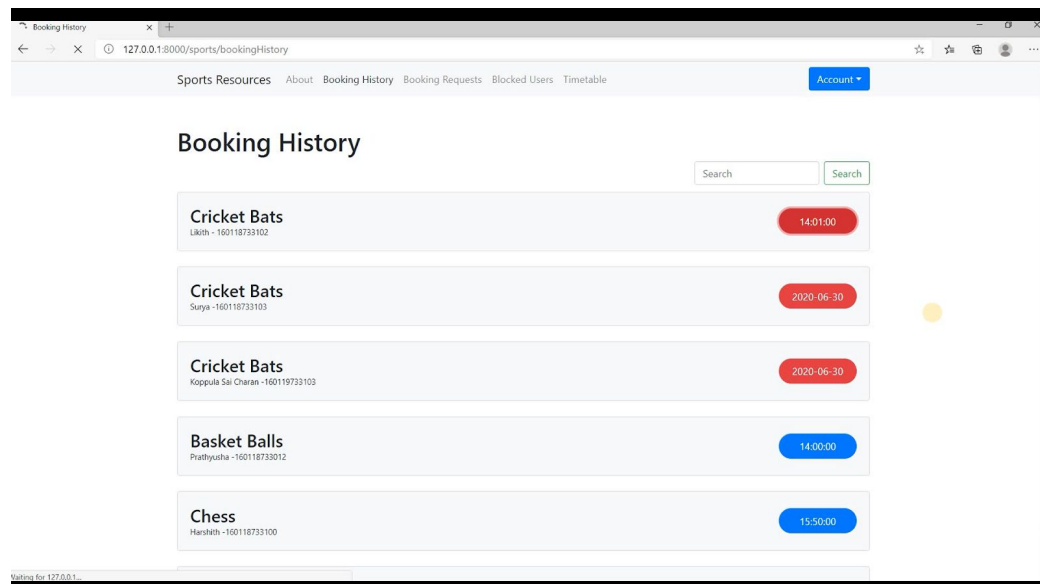
5.2.2.5 Foglalási kérelmek oldal):Ezen az oldalon a felhasználó által benyújtott foglalási kérések jelennek meg. A

Az adminisztrátor elfogadhatja vagy elutasíthatja a kérést.



23. ábra: Foglalási kérelmek oldal

5.2.2.6).Foglalási előzmények oldala):Ez az oldal az összes elfogadott foglalás előzményeit jeleníti meg. Ez az pirossal kiemelve, ha a felhasználó nem küldte vissza az erőforrást a megadott időn belül a kiadás időpontjának időbélyegzőjével együtt, és kékkel van kiemelve, ha a felhasználónak még van ideje visszaküldeni az erőforrást az időbélyeggel együtt, amikor erőforrás visszakerült.



24. ábra: Foglalási előzmények oldal

5.2.2.7 Letiltott felhasználók oldal: Ez az oldal a letiltott felhasználók listáját jeleníti meg. Arra is van lehetőség feloldja őket. Az adminisztrátor a plusz ikonra kattintva új felhasználókat is felvehet a listára.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/sports/blockedUsers'. The page title is 'Blocked users'. The navigation bar includes 'Sports Resources', 'About', 'Booking History', 'Booking Requests', 'Blocked Users', and 'Timetable'. A blue 'Account' button is on the right. The main content area is titled 'Blocked Users' and features a search bar with a 'Search' button. Below the search bar, there is a list of blocked users, each with a name, a sport, and an 'Unblock' button. The users listed are: 160117733115-varun (Foot Balls-fine50), 160118733101-Harish (Cricket Bats-fine50), 160119733082-Amith (Foot Balls-fine50), and 160118733102-Likith (Cricket Bats-fine50). A green plus icon is located at the bottom right of the page.

Name	Sport	Action
160117733115-varun	Foot Balls-fine50	Unblock
160118733101-Harish	Cricket Bats-fine50	Unblock
160119733082-Amith	Foot Balls-fine50	Unblock
160118733102-Likith	Cricket Bats-fine50	Unblock

+

The screenshot shows the same 'Blocked Users' page, but with a modal form open on the right side. The modal form has two input fields: 'Id' and 'Fine'. The 'Id' field contains the value '160118733012'. The 'Fine' field contains the value '300'. Below the input fields are two buttons: 'Submit' (green) and 'Close' (red). The list of blocked users is updated to include the new user: 160117733116-kedar (Footballs-fine50), 160117733115-varun (Footballs-fine50), 160117733001-Akshitha (Footballs-fine50), and 160118733011-Annanya (Cricket balls-fine500).

Name	Sport	Action
160117733116-kedar	Footballs-fine50	Unblock
160117733115-varun	Footballs-fine50	Unblock
160117733001-Akshitha	Footballs-fine50	Unblock
160118733011-Annanya	Cricket balls-fine500	Unblock

Id

160118733012

Fine

300

Submit

Close

25. és 26. ábra: Letiltott felhasználók oldal

5.2.2.8 Menetrend oldal:) Ez az oldal egy osztály órarendjét jeleníti meg a megadott bemenetek alapján.

Time table

Branch Year Section

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)							
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING							
Even Semester Time Table 2019-20							
w.e.f.: 30-03-2020							
Class: BE CSE2 IV SEM							
Period	I	II	III	12:15-1:15	IV	V	VI
Day	9:10-10:10	10:10-11:10	11:15-12:15		1:15-2:15	2:15-3:15	3:20-4:20
MON	IT WORKSHOP@CSE Lab8				DBMS	BE	
TUE	BE LAB-B1, CAMP LAB-B2@CSE Lab5, DBMS LAB-B3@CSE Lab3			L	DBMS	IC	BE
WED	DBMS	CAMP	PS	U	PS		
THU	IC	CAMP	PS	N	BE LAB-B2, CAMP LAB-B3@CSE Lab5, DBMS LAB-B1@CSE Lab3		
FRI	BE	PS	CAMP	C	BE LAB-B3, CAMP LAB-B1@CSE Lab5, DBMS LAB-B2@CSE Lab3		
SAT				H			

27. ábra: Menetrend oldal

6. TESZTELÉS ÉS ÉRVÉNYESÍTÉS

6.1 Bevezetés

A kód tesztelése és érvényesítése a projekt kulcsfontosságú része. Lehetetlen meghatározni az összes kivételes esetet, amely a tervezés során felmerülhet. Ezért a tesztelés hasznos módja annak, hogy megtaláljuk a hibákat mind a kód szerkezetében, mind a különböző funkciók megvalósításában.

6.2 Tesztesetek és forgatókönyvek tervezése

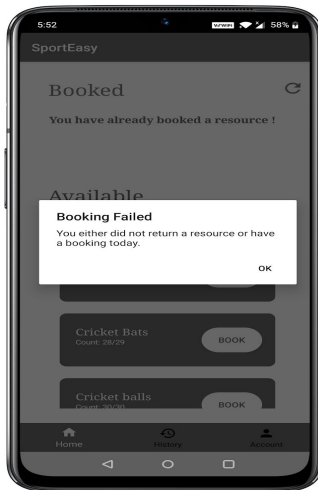
A teszteseteket úgy tervezték meg, hogy szem előtt tartsák azokat a helyeket, ahol a kód meghibásodhat, és hibákat dobhat vagy hibás kimeneteket ad. A következő esetek néhány fontos forgatókönyv, amelyet szem előtt kell tartani.

6.2.1 Elfelejtett jelszó:Ha a felhasználó elfelejtette a jelszavát, e-mailt kap a visszaállításról jelszó.



28. ábra: Elfelejtett jelszó

6.2.2 A bírsággal sújtott felhasználó foglalási kérelme: Ha a felhasználó annak ellenére megpróbálja lefoglalni az erőforrást jól van, akkor megjelenik egy üzenet, hogy a felhasználó jól van.



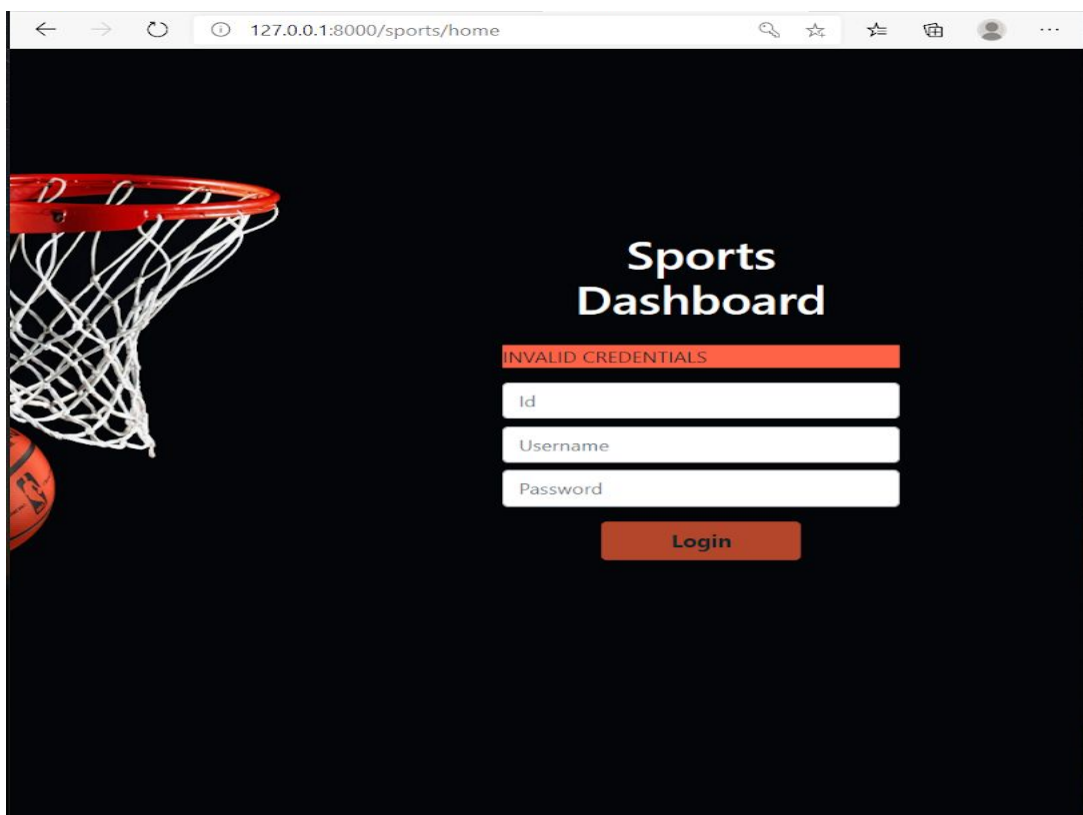
29. ábra: Foglalási kérelmet a felhasználó bírsággal

6.2.3 Jelszó módosítása: Ha a felhasználó meg akarja változtatni a jelszavát, akkor ezt a belépéssel megteheti új jelszót és megerősítést.



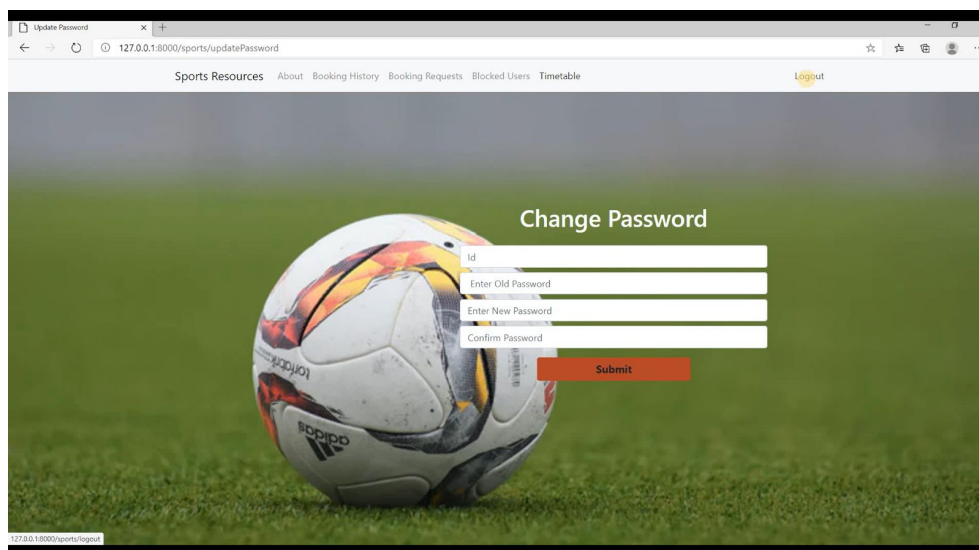
30. ábra: Felhasználó jelszavának módosítása

6.2.4 Érvénytelen hitelesítő adatok:Ha az adminisztrátor rossz hitelesítési adatokat próbál megadni, megjelenik egy üzenet, amely az Érvénytelen hitelesítési adatokat jelzi.



31. ábra: Az adminisztrátor érvénytelen hitelesítő adatai

6.2.5 Jelszó módosítása:Ha az admin meg akarja változtatni a jelszót, akkor ezt megteheti új jelszó megadása és megerősítése.

The image shows a web browser window with the address bar displaying '127.0.0.1:8000/sports/updatePassword'. The page has a navigation bar with links: 'Sports Resources', 'About', 'Booking History', 'Booking Requests', 'Blocked Users', and 'Timetable'. A 'Logout' button is in the top right. The main content area features a background image of a soccer ball on a green field. Overlaid on this is a 'Change Password' form with the following fields: 'Id', 'Enter Old Password', 'Enter New Password', and 'Confirm Password'. Below these fields is an orange 'Submit' button. The browser's developer tools are visible at the bottom left, showing the URL '127.0.0.1:8000/sports/updatePassword'.

31. ábra: Jelszó módosítása az admin számára

6.3 Érvényesítés

A kód érvényesítése a fent említett összes tesztet futtatásával történt, például rossz és helyes bejelentkezési adatokkal. Ügyeltünk arra, hogy minden lehetséges kivételes esetet és egyéb problémás forgatókönyvet belefoglaljanak a bemenetbe.

7. KÖVETKEZTETÉS

7.1 Következtetés

Ez a projekt a sporteszköz kölcsönzéséhez alkalmazott munka számítógépesítésére szolgál. Ez egy nagy előrelépés a kézi rendszerhez képest. A rendszer számítógépesítése felgyorsította a folyamatot. A jelenlegi rendszerben a front office menedzsment nagyon lassú. Ez az alkalmazás volt áladatokkal alaposan ellenőrizték, és így nagyon megbízhatónak találták. A szoftver veszi minden követelménynek megfelel, és képes egyszerű és hatékony tárolási információkat nyújtani sportforrásokat foglaló diákokkal kapcsolatos. Reszponzív webdesign és mobilalkalmazások minden területen könnyebben és biztonságosabban dolgozhat. Könnyen elérhető felhasználói felület biztosításával a Az erőforrások lefoglalásának és jóváhagyásának teljes folyamata, valamint más funkciók is lettek rendkívül egyszerű, miközben jelentősen kiküszöböli az egyébként előforduló számos hibát kézi rendszerben okozzák. Ezzel a platformmal, amit fejlesztettünk, reméljük, hogy a következő:

- Csökkentett idővesztés
- Kényelmes létesítmények biztosítása a tanulók számára
- Könnyű adatáramlást biztosít
- Kevesebb munkavállalói befektetés

7.2 Korlátozások

A diákok számára kifejlesztett mobilalkalmazás sportforrások lefoglalására az csak az Android 5.0+ verzióival kompatibilis, minimális androidos futásidejű verzió szükséges hozzá 5.0

Az adminisztrátornak ellenőriznie kell a menetrendet az erőforrás kiadása közben.

A felhasználónak nincs határideje az esedékesség törlésére, és nincs értesítési rendszer a felhasználó intimizálására az esedékes törléséhez.

7.3 Jövőbeli hatály

Ez az alkalmazás számos új funkcióval bővíthető. Az egyik ilyen funkció az értesítési rendszer hozzáadása, amelyen keresztül az adminisztrátor értesítést küldhet a felhasználónak a foglalásait. A jelenlegi rendszerben a felhasználónak manuálisan kell belépnie az alkalmazásba, és ellenőriznie kell, hogy van-e az erőforrást elfogadják vagy sem. Ehelyett ennek javítása egy visszaigazolás küldése lenne a felhasználónak, amikor foglalási kérelmét az adminisztrátor jóváhagyta. Elérhetetlensége esetén a erőforrás, egy „értesíts, ha elérhető” gomb is beilleszthető. Jelenleg, amikor egy diák erőforrást szeretne lefoglalni, akkor az adminisztrátor manuálisan ellenőrzi, hogy a hallgatónak van-e szabad órája adott időpontban, és ennek megfelelően adja ki az erőforrást. De a jövőben lehet olyan alkalmazást készíteni, amely maga észleli, hogy a tanulónak van-e szabad órája vagy sem, ezáltal kiküszöböli a az admin bevonása. Ezt követően egy külön oldalt is létrehozhat, amely az admin oldalhoz kapcsolódik kifejezített, ahol az adminisztrátor frissítéseket adhat hozzá a közelgő sporteseményekről vagy versenyekről így ezek a frissítések a hallgatók mobilalkalmazásában is láthatóak lesznek

lehetővé teszi számukra, hogy a mobiljukon keresztül értesüljenek az összes folyamatban lévő vagy közelgő sporteseményről.

IRODALOM

1. <https://www.javatpoint.com/java-tutorial>
2. <https://www.quora.com>
3. <https://www.javatpoint.com/android-tutorial>
4. <https://www.tutorialspoint.com/java/index.htm>
5. <https://www.w3resource.com/java-exercises/>
6. <https://www.wikipedia.com>
7. <https://www.google.com/django-documentation/>
8. <https://www.programiz.com/java-programming>
9. <https://www.tutorialspoint.com/android/index.htm>
10. <https://developer.android.com/guide>
11. <https://developer.android.com/training/basics/firstapp>
12. <https://www.vogella.com/tutorials/android.html>
13. <https://www.youtube.com/watch?v=jEmq1B1gveM>
14. <https://firebase.google.com/docs/android/setup>
15. <https://www.w3schools.com/java/>
16. <https://www.tutorialspoint.com/flask>

FÜGGELÉK - I

<https://github.com/indooriaditi/cosc4-webteam> <https://github.com/amith2610/Android-Team4> <https://github.com/gsaiprathyusha/apiteam04> <https://sport-resources-booking-api.herokuapp.com>