

JEGYZŐKÖNYV

Web-Technológiák II.

Féléves feladat

Webes könyvtári nyilvántartórendszer létrehozása

Készítette: **Bartók-Balog Péter**
Neptunkód: **ASNQPH**

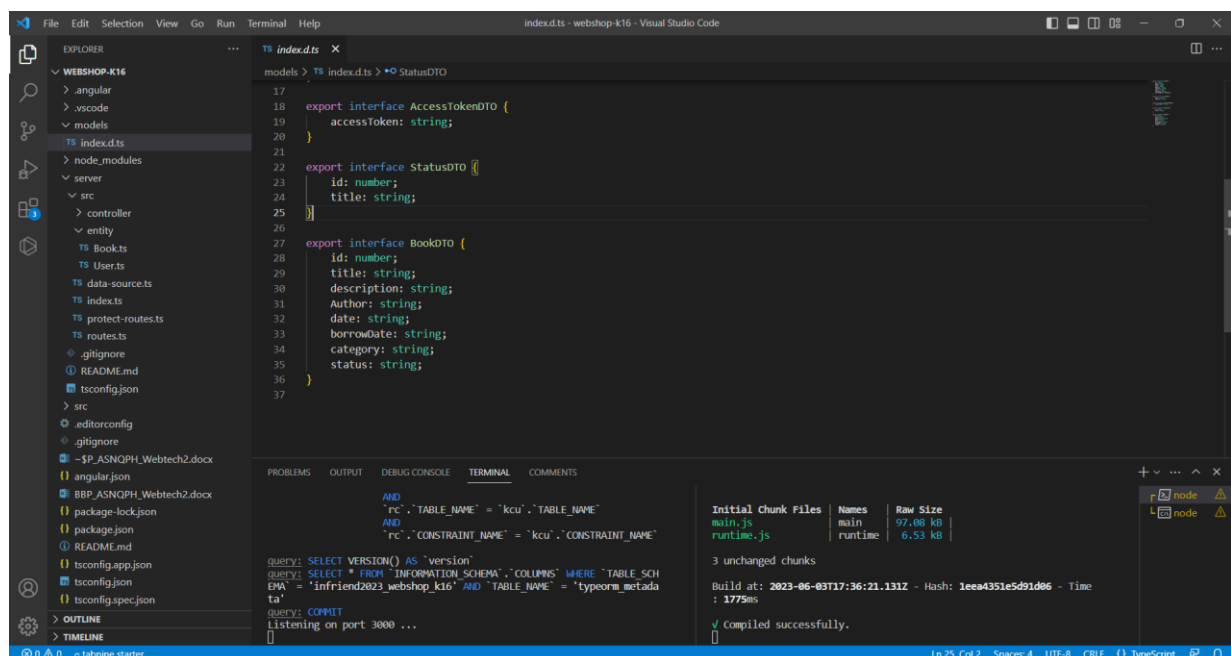
A feladat leírása:

A feladat egy weboldal létrehozása Angular, Node.js, MongoDB technológiát alkalmazva, a megadott feltételek alapján. A feladat megoldásához létre kell hozni egy bejelentkező felületet, illetve a regisztráció, és a bejelentkezés után, létre kell hozni egy valamilyen termék nyilvántartására alkalmas kezelőfelületet. A feladat megoldása kezdődik a backend létrehozásával, majd a frontend megalkotásával. A feladat teljesítéséhez szükséges egy dokumentáció, amiben bemutatásra kerül a felhasználói felület és a megírt kód.

Kód, és a felhasználói felület bemutatása:

Backend:

Kezdetben létrehozzuk a megfelelő fájlszerkezetet a TypeORM részére és létrehozzuk a kezdeti identitásainkat(táblákat) melyek adatait majd később fel akarjuk használni. Ahhoz, hogy ezekkel a táblákkal biztonságban tudjunk dolgozni, létrehozunk egy velük megegyező szerkező DTO gyűjteményt is, mellyel majd fogadunk, és továbbítunk adatokat a frontend és backend között. Ezalatt a munkafázis alatt módosítjuk az alapvető adatait az alkalmazásunknak, (webnév, importok, létrehozó, linkek stb.)



Ezt követően létrehozzuk a controllereket, azaz az entitásokhoz kapcsolódó kezelőmodulokat, melyek majd a táblákkal történő alapvető, és táblaspecifikus műveleteket fogják kezelni. Ilyen művelet az id alapján lekérdezés, a listázás, törlés, módosítás, létrehozás, számolás, kölcsönzés stb.

```
server > src > controller > TS base.controllers.ts > Controller > borrowBook
1 import { Repository } from 'typeorm';
2 import { AppDataSource } from '../data-source';
3 import { Book } from '../entity/book';
4
5 export abstract class Controller {
6     repository: Repository<any>;
7
8     getAll = async (req, res) => {
9         try {
10             const entities = await this.repository.find();
11             res.json(entities);
12         } catch (err) {
13             this.handleError(res, err);
14         }
15     };
16
17     getOne = async (req, res) => {
18         try {
19             const id = req.params.id;
20             const entity = await this.repository.findOneBy({ id: id });
21             if (!entity) {
22                 return this.handleError(res, null, 404, 'Not found.');

| Initial Chunk Files | Names   | Raw Size |
|---------------------|---------|----------|
| main.js             | main    | 97.08 kB |
| runtime.js          | runtime | 6.53 kB  |



3 unchanged chunks



Build at: 2023-06-03T17:36:21.131Z - Hash: 1eea4351e5d91d06 - Time: 1779ms



✓ Compiled successfully.


```

Ezt követően be kell állítanunk az útvonalakat, melyeken keresztül a felhasználó el tudja érni ezeket a kezelőket, és ténylegesen tud végrehajtani módosításokat a szerver oldalunkon. Ehhez háromféle metódust használunk: GET, POST, és a DELETE a linkekhez igazítva.

```
server > src > TS routes.ts > getRoutes
1 import { UserController } from './controller/user.controller';
2 import { checkUser } from './protect-routes';
3
4 export function getRoutes() {
5     const router = express.Router();
6
7     const bookController = new BookController();
8     router.get('/books', bookController.getAll);
9     router.post('/books', checkUser, bookController.create);
10    router.put('/books', checkUser, bookController.update);
11    router.get('/books/available', bookController.getAvailableBooks);
12    router.post('/books/borrow', checkUser, bookController.borrowBook);
13    router.post('/books/return', bookController.returnBook);
14    router.get('/books/overdue', bookController.getOverdueBooks);
15    router.get('/books/:id', bookController.getOne);
16    router.delete('/books/:id', checkUser, bookController.delete);
17
18    const userController = new UserController();
19    router.get('/users', userController.getAll);
20    router.get('/users/:id', userController.getOne);
21    router.post('/users', userController.create);
22    router.put('/users', checkUser, userController.update);
23    router.delete('/users/:id', checkUser, userController.delete);
24    router.post('/users/login', userController.login);
25
26    return router;
27 }
```

Initial Chunk Files	Names	Raw Size
main.js	main	97.08 kB
runtime.js	runtime	6.53 kB

3 unchanged chunks

Build at: 2023-06-03T17:36:21.131Z - Hash: 1eea4351e5d91d06 - Time: 1779ms

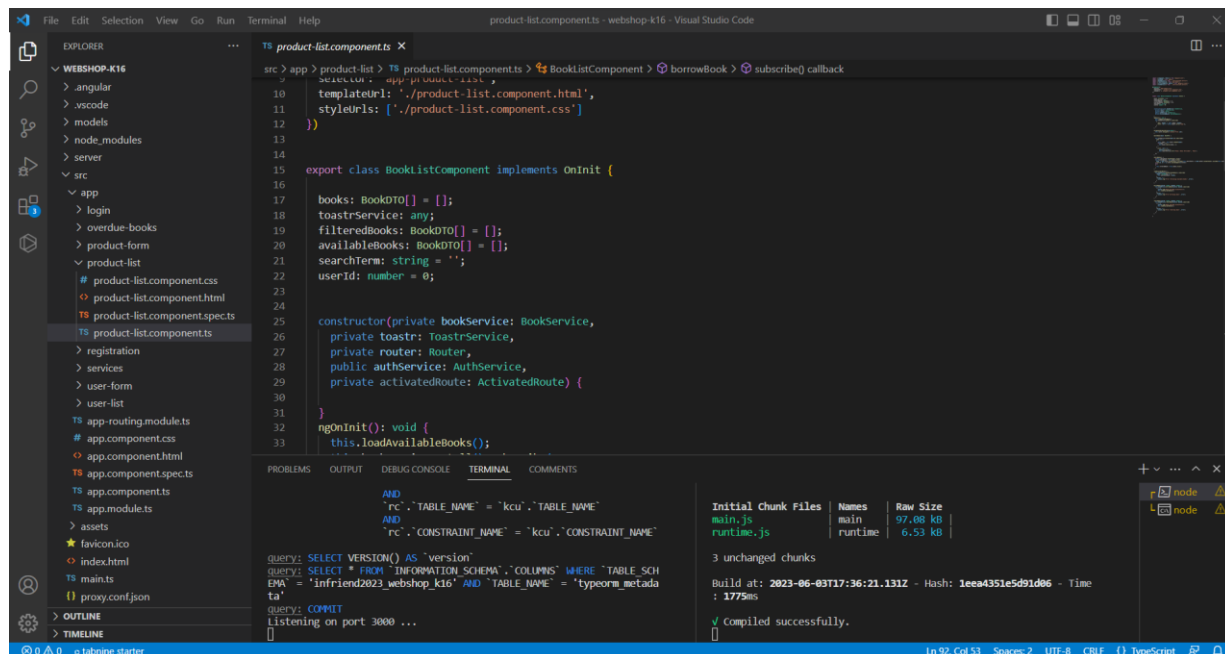
✓ Compiled successfully.

Frontend:

Amint a backend készen áll, folytatjuk a munkát a kliens oldallal, tehát azzal, amit a felhasználó fog fizikailag kezelni, és használni.

Minden fontosabb oldalhoz, kezelőhöz létrehozunk egy-egy komponens a parancssoron keresztül, mely tartalmaz kettő TS, egy HTML, illetve egy formázáshoz szükséges CSS fájlt. (ng g c 'komponensnev')

Elsősorban megírjuk a ts fájlban a komponens műveleteit, melyet majd a felhasználó a weboldal kezelésére fog használni, Behúzzuk a megfelelő importokat, példányosítunk, illetve inicializáljuk a változókat.



```
src > app > product-list > TS product-list.component.ts > TS BookListComponent > @borrowBook > @subscribe() callback
10 templateUrl: './product-list.component.html',
11 styleUrls: ['./product-list.component.css']
12 })
13
14
15 export class BookListComponent implements OnInit {
16
17   books: BookDTO[] = [];
18   toastrService: any;
19   filteredBooks: BookDTO[] = [];
20   availableBooks: BookDTO[] = [];
21   searchTerm: string = '';
22   userId: number = 0;
23
24
25   constructor(private bookService: BookService,
26               private toastr: ToastrService,
27               private router: Router,
28               public authService: AuthService,
29               private activatedRoute: ActivatedRoute) {
30
31   }
32   ngOnInit(): void {
33     this.loadAvailableBooks();
34   }
35 }
```

```
query: SELECT VERSION() AS "version"
query: SELECT * FROM "INFORMATION_SCHEMA"."COLUMNS" WHERE "TABLE_SCH
EMA" = 'infriend2023_webshop_k16' AND "TABLE_NAME" = 'typeorm_meta
ta'
query: COMMIT
Listening on port 3000 ...
```

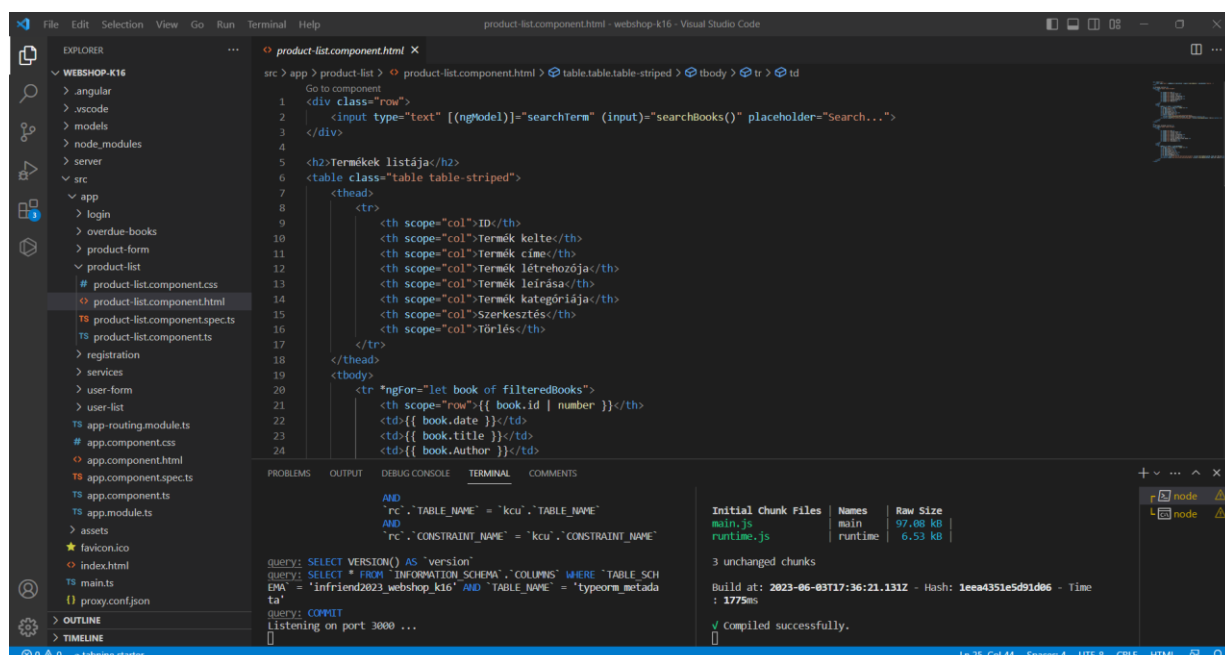
Initial Chunk Files	Names	Raw Size
main.js	main	97.08 kB
runtime.js	runtime	6.53 kB

3 unchanged chunks

Build at: 2023-06-03T17:36:21.131Z - Hash: 1eea4351e5d91d06 - Time: 1775ms

Compiled successfully.

Amint ezzel megvagyunk létrehozzuk a fizikális kezelőfelület a HTML fájlban keresztül. Jelen esetben alkalmazzuk a webtechnológiák 1.-ben tanultakat és táblázatokat felhasználva listázzuk ki a könyveket, írjuk meg a kereső mezőt, hozzuk létre a felületet.



```
src > app > product-list > product-list.component.html > table.table-striped > tbody > tr > td
Go to component
1 <div class="row">
2   <input type="text" [(ngModel)]="searchTerm" (input)="searchBooks()" placeholder="Search..."
3 </div>
4
5 <h2>Termékek listája</h2>
6 <table class="table table-striped">
7   <thead>
8     <tr>
9       <th scope="col">ID</th>
10      <th scope="col">Termék neve</th>
11      <th scope="col">Termék címe</th>
12      <th scope="col">Termék létrehozója</th>
13      <th scope="col">Termék leírása</th>
14      <th scope="col">Termék kategóriája</th>
15      <th scope="col">Szerkesztés</th>
16      <th scope="col">Törölés</th>
17    </tr>
18  </thead>
19  <tbody>
20    <tr *ngFor="let book of filteredBooks">
21      <th scope="row">{{ book.id | number }}</th>
22      <td>{{ book.date }}</td>
23      <td>{{ book.title }}</td>
24      <td>{{ book.Author }}</td>
```

```
query: SELECT VERSION() AS "version"
query: SELECT * FROM "INFORMATION_SCHEMA"."COLUMNS" WHERE "TABLE_SCH
EMA" = 'infriend2023_webshop_k16' AND "TABLE_NAME" = 'typeorm_meta
ta'
query: COMMIT
Listening on port 3000 ...
```

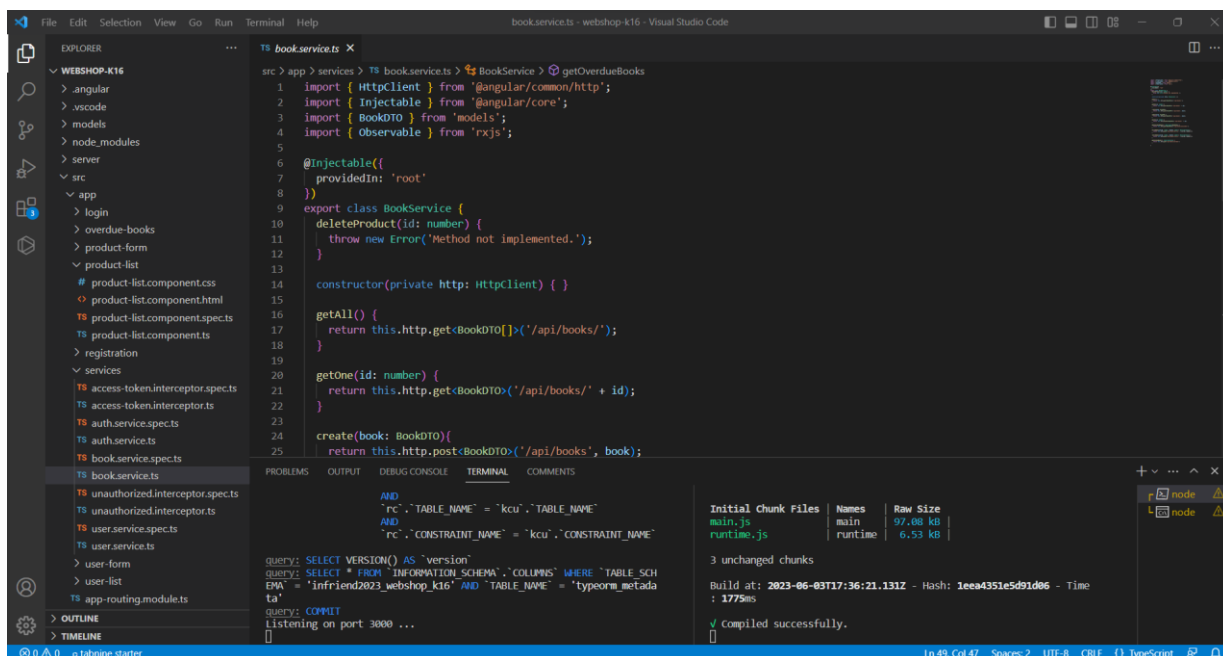
Initial Chunk Files	Names	Raw Size
main.js	main	97.08 kB
runtime.js	runtime	6.53 kB

3 unchanged chunks

Build at: 2023-06-03T17:36:21.131Z - Hash: 1eea4351e5d91d06 - Time: 1775ms

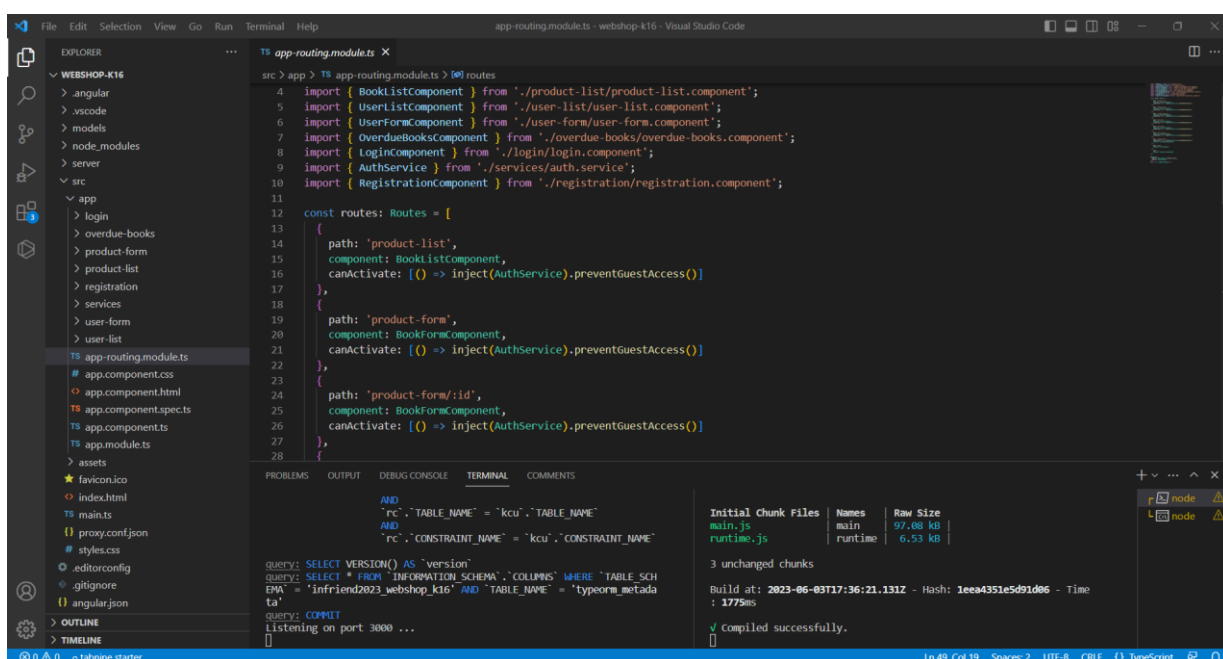
Compiled successfully.

Ahhoz, hogy ezek rendesen működjenek, létre kell hoznunk a szervíz réteget, minden egyes entitáshoz, komponenshez, hogy a kliens oldalról érkező kéréseket összekössük a szerver oldallal, hogy a módosítások, lekérdezések, metódusok megfelelően működjenek.



Ennek megfelelően lépésről lépésre hozzuk létre az összes szükséges komponens, és azok metódusait a megfelelő szervízzel összekötve.

Amint minden komponens rendelkezésünkre áll, beállíthatjuk azok útvonalatait, azok megfelelő komponenseivel együtt, és a nyilvántartásunk készen áll.



A weboldal képes a következőkre:

- Regisztráció
- Bejelentkezés (Titkosítással)
- Kijelentkezés
- Tételek, felhasználók listázása
- Tételek, felhasználók módosítása
- Tételek, felhasználók törlése
- Új tétel, felhasználó létrehozása
- Keresés paraméter alapján
- Tételek kölcsönzése

The screenshot shows a web browser window with the address bar set to localhost:4200. The browser's tab bar includes several open tabs, and the address bar shows a search bar with the text "További könyvtárak". The main content area displays the "Könyvtári Nyilvántartó Rendszer" (Library Management System) interface. The top navigation bar contains links: "Könyvtári Nyilvántartó Rendszer", "Könyvek", "Új könyv", "Felhasználók", "Új felhasználó", "Késésben lévő könyvek", and "Kilépés". The "Regisztráció" (Registration) form is active, featuring input fields for "Név" (Name), "Telefonszám" (Phone number), "Személyi igazolvány szám" (Personal ID number), "Lakcím" (Address), "E-mail cím" (Email address), and "Jelszó" (Password). The "Név" field is pre-filled with "Bartók-Balog Péter", "Telefonszám" with "7133", "Személyi igazolvány szám" with "52525", "Lakcím" with "Deák Ferenc tér", and "E-mail cím" with "valami@gmail.com". The "Jelszó" field is masked with six dots. A "Regisztráció" button is located below the form. Below the registration form, the "Kilépés" (Logout) link is visible. The "Bejelentkezés" (Login) form is also visible, featuring input fields for "E-mail cím" and "Jelszó", and a "Belépés" (Login) button.

Könyvtári Nyilvántartó Rendszer Könyvek Új könyv Felhasználók Új felhasználó Késésben lévő könyvek Kilépés

Regisztráció

Név
Bartók-Balog Péter

Telefonszám
7133

Személyi igazolvány szám
52525

Lakcím
Deák Ferenc tér

E-mail cím
valami@gmail.com

Jelszó

Regisztráció

Könyvtári Nyilvántartó Rendszer Könyvek Felhasználók Késésben lévő könyvek Kilépés

E-mail cím

Jelszó

Belépés

Search...

Termékek listája

ID	Termék kelte	Termék címe	Termék létrehozója	Termék leírása	Termék kategóriája	Szerkesztés	Törlés
15	2023-05-31	semmi	senki	sdad	kazetta	<button>Szerkesztés</button>	<button>Törlés</button>

Elérhető termékek listája

ID	Termék kelte	Termék címe	Termék létrehozója	Termék leírása	Termék kategóriája	Termék kölcsönzésének ideje	Kölcsönzés	Visszahozás
15	2023-05-31	semmi	senki	sdad	kazetta		<button>Borrow</button>	<button>Return</button>

Könyv adatok

Cím

valami

Szerző

valaki

Leírás

valami

Kiadás éve

2023. 06. 03.

Kategória [cd,könyv,kazetta]

CD

Mentés