



# RSVTI HELPER

[RSVTI - RESPONSABIL CU  
SUPRAVEGHEREA ȘI VERIFICAREA  
TEHNICĂ A INSTALAȚIILOR]

---

BACIU DAN RAUL

# SCOPUL PROIECTULUI

Pentru desfășurarea activității un RSVTI are nevoie de:

1. ○ bază de date în care să se stocheze informațiile despre angajații și instalațiile din cadrul companiilor supravegheate.
2. ○ interfață prin care să se poată:
  - vizualiza datele angajaților și al instalațiilor
  - să poată adăuga sau șterge companii, angajați și instalații noi
  - să poată edita datele acestora
  - să poată autoriza angajații în diferite calificări din domeniul ISCIR (Inspectia de Stat pentru Controlul Cazanelor, Recipientelor sub Presiune si Instalatiilor de Ridicat) sau să adauge autorizații deținute de către angajați
  - să poată autoriza instalațiile sau să adauge autorizații date de către CNCIR
  - să poată scoate procese verbale, adeverințe sau rapoarte despre angajați sau instalații

Aplicația a fost dezvoltată utilizând pentru:

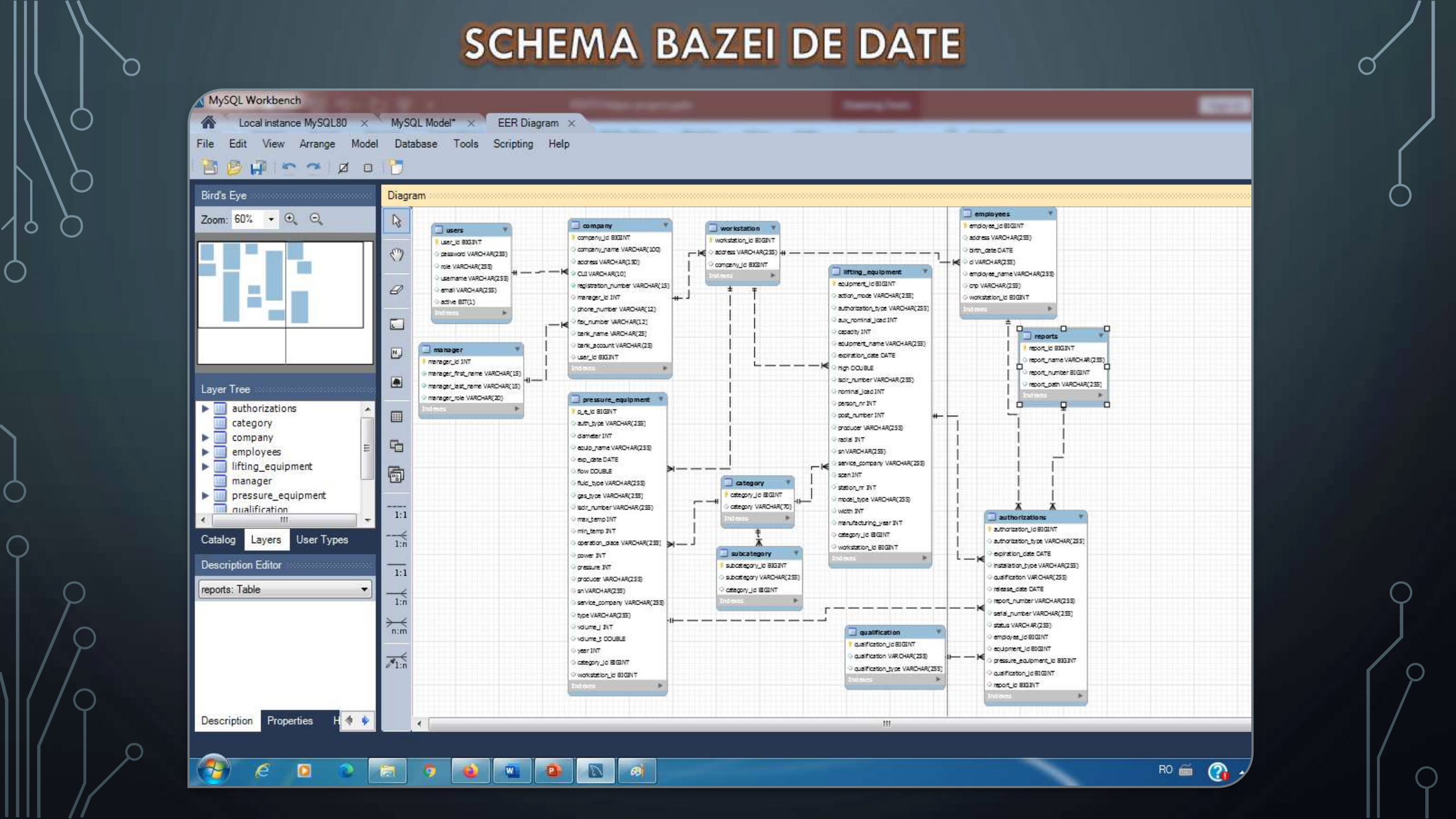
1. **Baza de date : MySQL**
2. **Back-End : Java - framework-ul Spring**
3. **Front-End : TypeScript - framework-ul Angular**

# SCHEMA BAZEI DE DATE

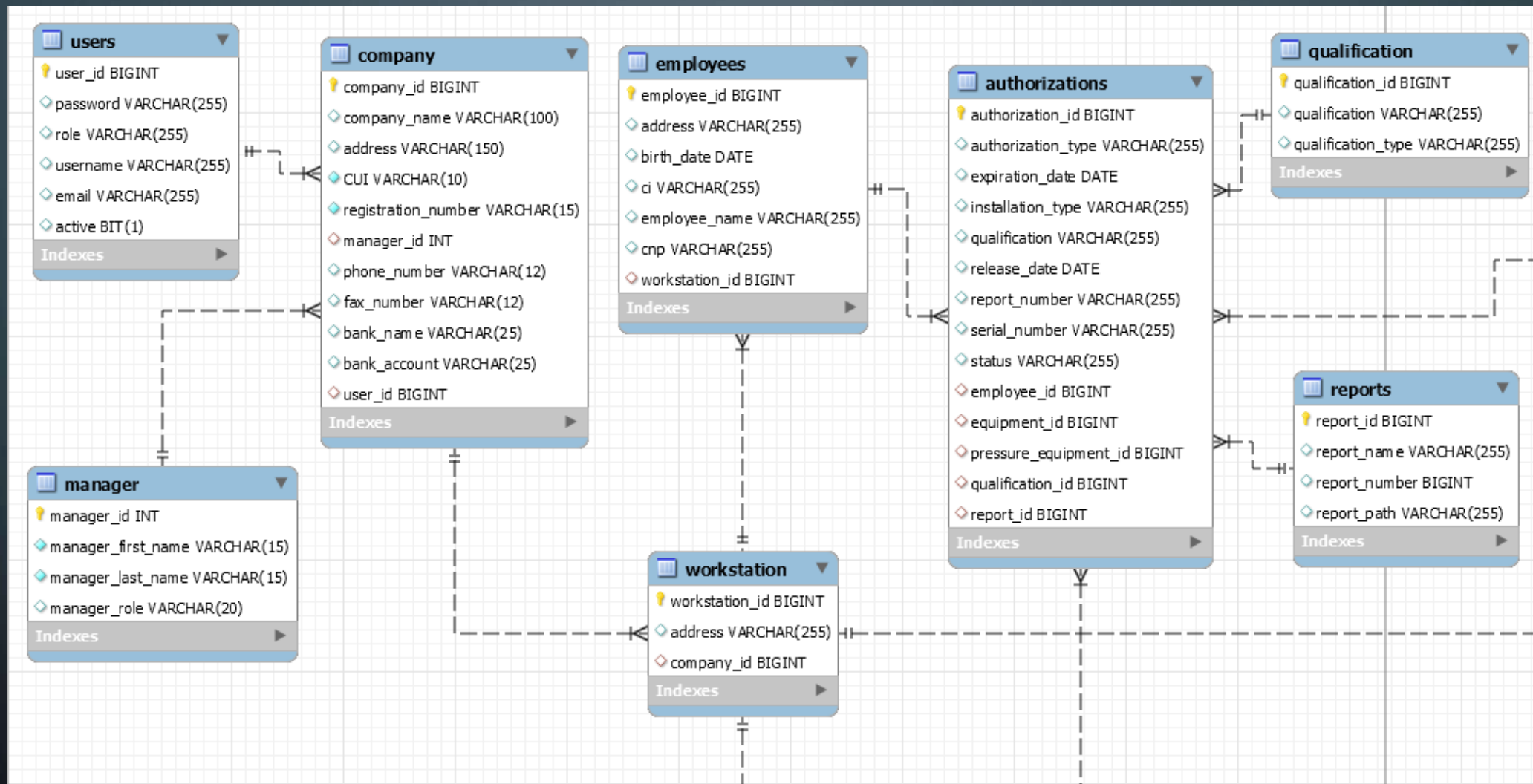
The screenshot displays the MySQL Workbench interface with an Entity-Relationship (EER) diagram. The diagram shows the following entities and their attributes:

- users**: user\_id BIGINT, password VARCHAR(255), role VARCHAR(255), username VARCHAR(255), email VARCHAR(255), active BIT(1).
- company**: company\_id BIGINT, company\_name VARCHAR(100), address VARCHAR(150), CUI VARCHAR(10), registration\_number VARCHAR(15), manager\_id INT, phone\_number VARCHAR(12), fax\_number VARCHAR(12), bank\_name VARCHAR(25), bank\_account VARCHAR(25), user\_id BIGINT.
- workstation**: workstation\_id BIGINT, address VARCHAR(255), company\_id BIGINT.
- lifting\_equipment**: equipment\_id BIGINT, action\_mode VARCHAR(255), authorization\_type VARCHAR(255), aux\_nominal\_load INT, capacity INT, equipment\_name VARCHAR(255), expiration\_date DATE, high DOUBLE, ldr\_number VARCHAR(255), nominal\_load INT, person\_lnr INT, post\_number INT, producer VARCHAR(255), radii INT, sn VARCHAR(255), service\_company VARCHAR(255), serial INT, station\_lnr INT, mode\_type VARCHAR(255), width INT, manufacturing\_year INT, category\_id BIGINT, workstation\_id BIGINT.
- pressure\_equipment**: p\_e\_id BIGINT, auth\_type VARCHAR(255), diameter INT, equip\_name VARCHAR(255), exp\_date DATE, flow DOUBLE, fluid\_type VARCHAR(255), gas\_type VARCHAR(255), ldr\_number VARCHAR(255), max\_temp INT, min\_temp INT, operation\_date VARCHAR(255), power INT, pressure INT, producer VARCHAR(255), sn VARCHAR(255), service\_company VARCHAR(255), type VARCHAR(255), volume\_l INT, volume\_s DOUBLE, year INT, category\_id BIGINT, workstation\_id BIGINT.
- category**: category\_id BIGINT, category VARCHAR(70).
- subcategory**: subcategory\_id BIGINT, subcategory VARCHAR(255), category\_id BIGINT.
- employees**: employee\_id BIGINT, address VARCHAR(255), birth\_date DATE, c VARCHAR(255), employee\_name VARCHAR(255), emp VARCHAR(255), workstation\_id BIGINT.
- reports**: report\_id BIGINT, report\_name VARCHAR(255), report\_number BIGINT, report\_path VARCHAR(255).
- authorizations**: authorization\_id BIGINT, authorization\_type VARCHAR(255), expiration\_date DATE, installation\_type VARCHAR(255), qualification VARCHAR(255), release\_date DATE, report\_number VARCHAR(255), serial\_number VARCHAR(255), status VARCHAR(255), employee\_id BIGINT, equipment\_id BIGINT, pressure\_equipment\_id BIGINT, qualification\_id BIGINT, report\_id BIGINT.
- qualification**: qualification\_id BIGINT, qualification VARCHAR(255), qualification\_type VARCHAR(255).

The diagram illustrates relationships between these entities, including one-to-many and many-to-many connections, with various cardinalities (1:1, 1:n, n:m) and roles (manager, user, employee) indicated.



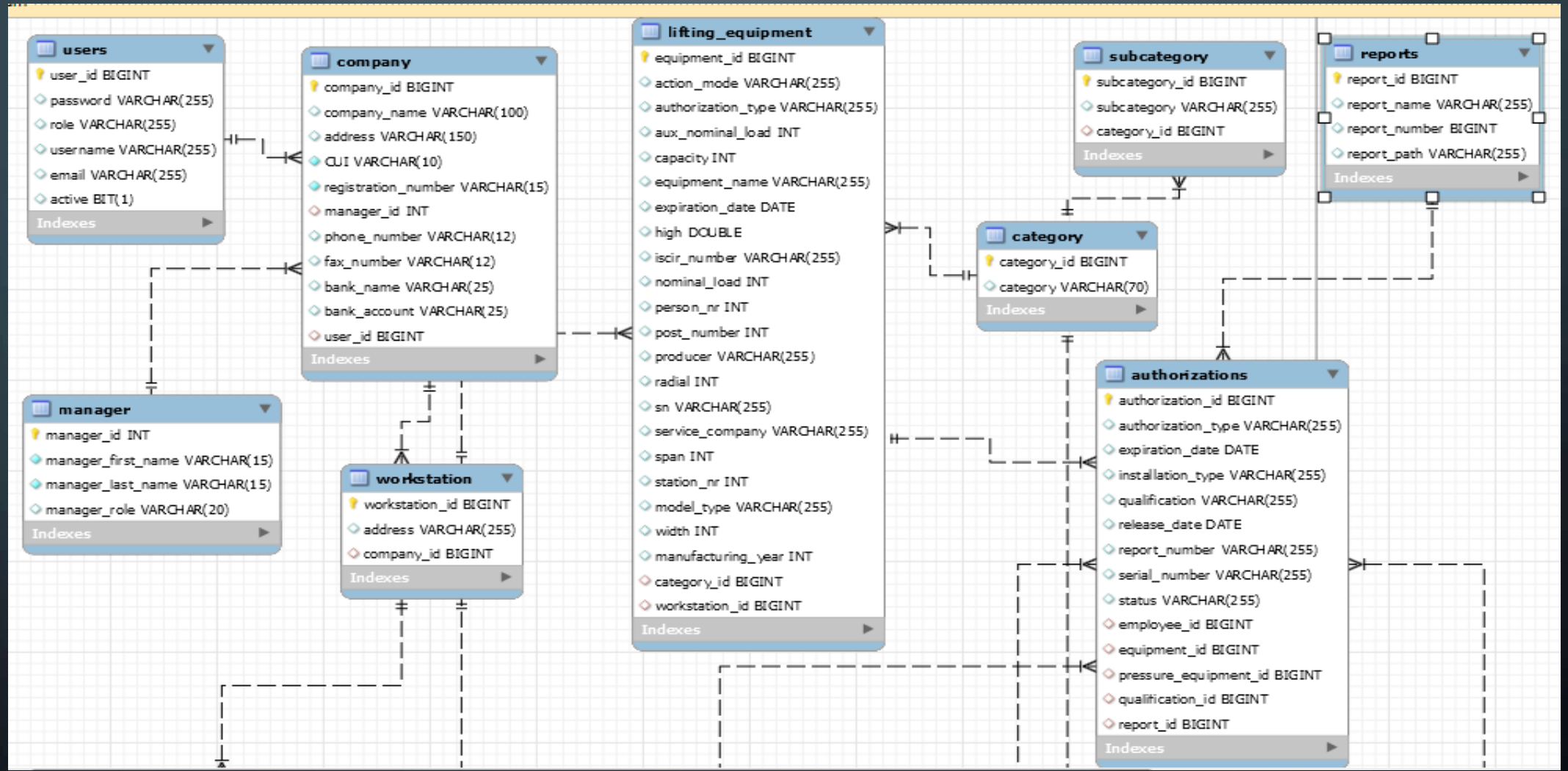
# SCHEMA BAZEI DE DATE



Companii -> Puncte de lucru -> Angajați -> Autorizații

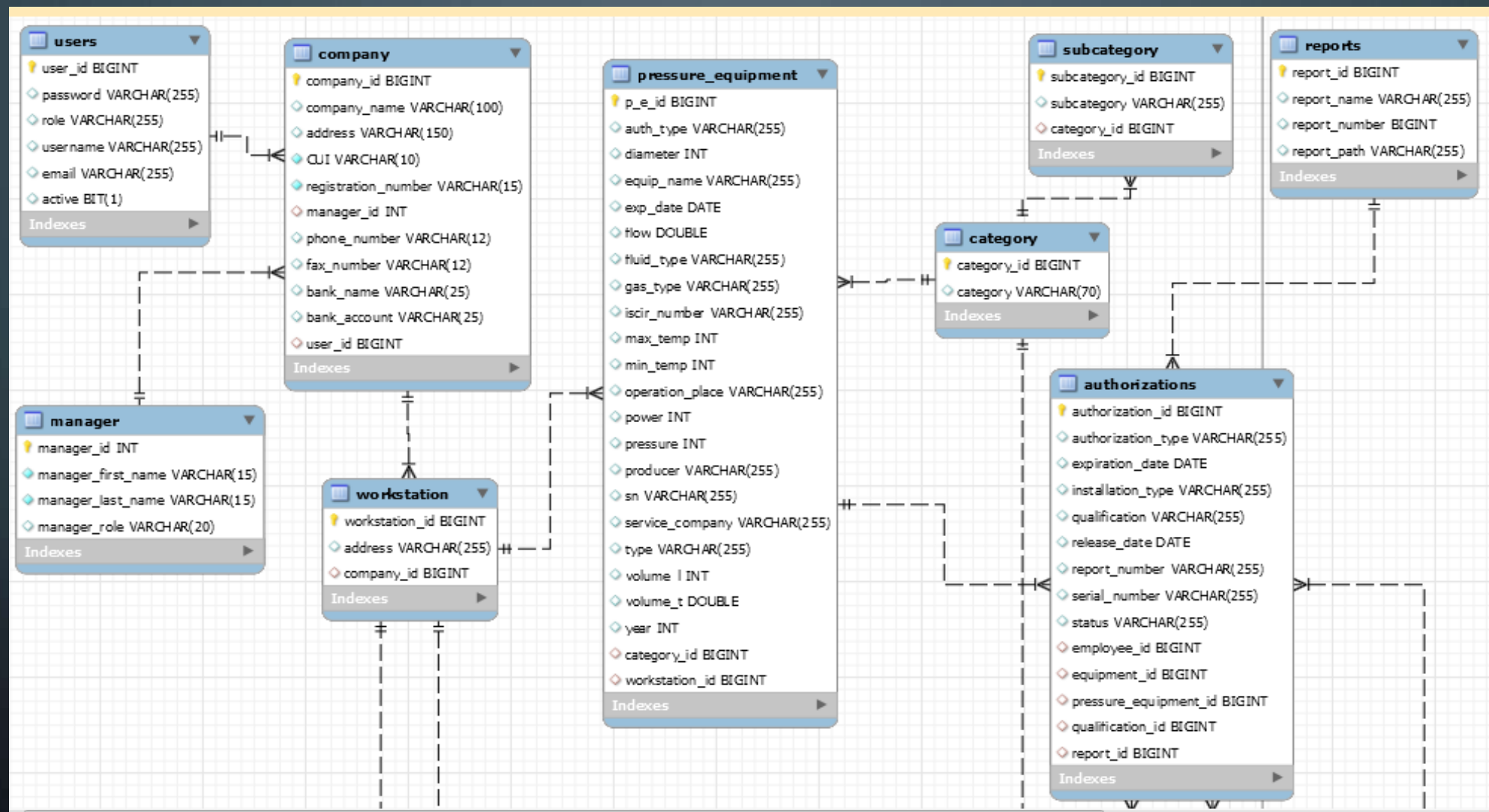


# SCHEMA BAZEI DE DATE



Companii -> Puncte de lucru -> Echipamente de ridicat -> Autorizații

# SCHEMA BAZEI DE DATE

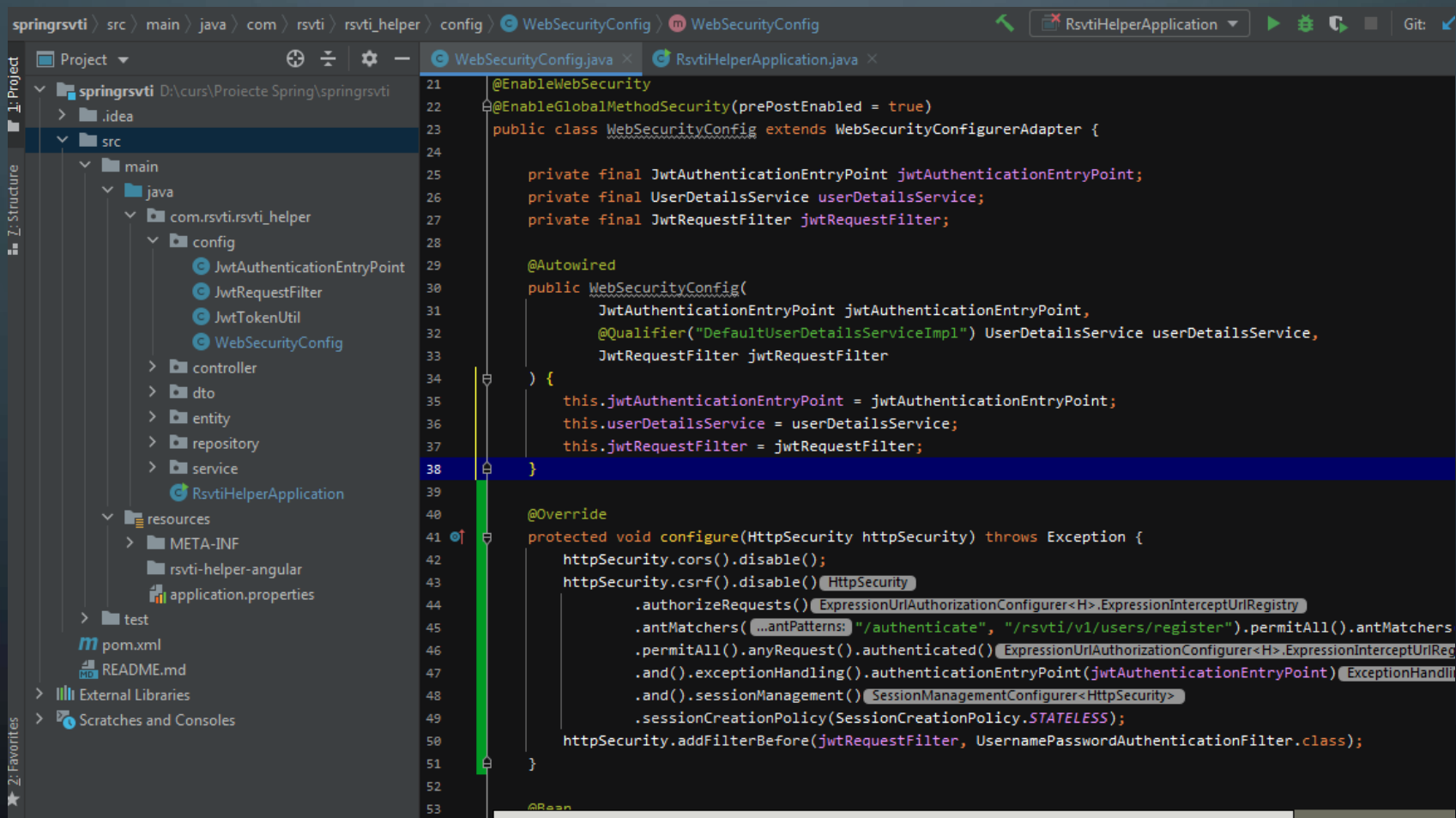


Companii -> Puncte de lucru -> Instalații sub presiune -> Autorizații

# STRUCTURA PROGRAMULUI

## Back-End : Spring

**Pachetul Config-** conține anumite configurații necesare aplicației, cum ar fi configurarea securității prin care intrarea în aplicație se face doar pe bază de autentificare



The screenshot displays an IDE interface with a project structure on the left and a code editor on the right. The project structure shows a package hierarchy: `src/main/java/com/rsvti/rsvti_helper/config`. The code editor shows the `WebSecurityConfig.java` file, which is a Spring Security configuration class. It extends `WebSecurityConfigurerAdapter` and is annotated with `@EnableWebSecurity` and `@EnableGlobalMethodSecurity(prePostEnabled = true)`. The class contains private fields for `JwtAuthenticationEntryPoint`, `UserDetailsService`, and `JwtRequestFilter`, which are injected using `@Autowired`. The `configure` method is overridden to configure `HttpSecurity`, including CORS, CSRF, request authorization, session management, and adding a request filter.

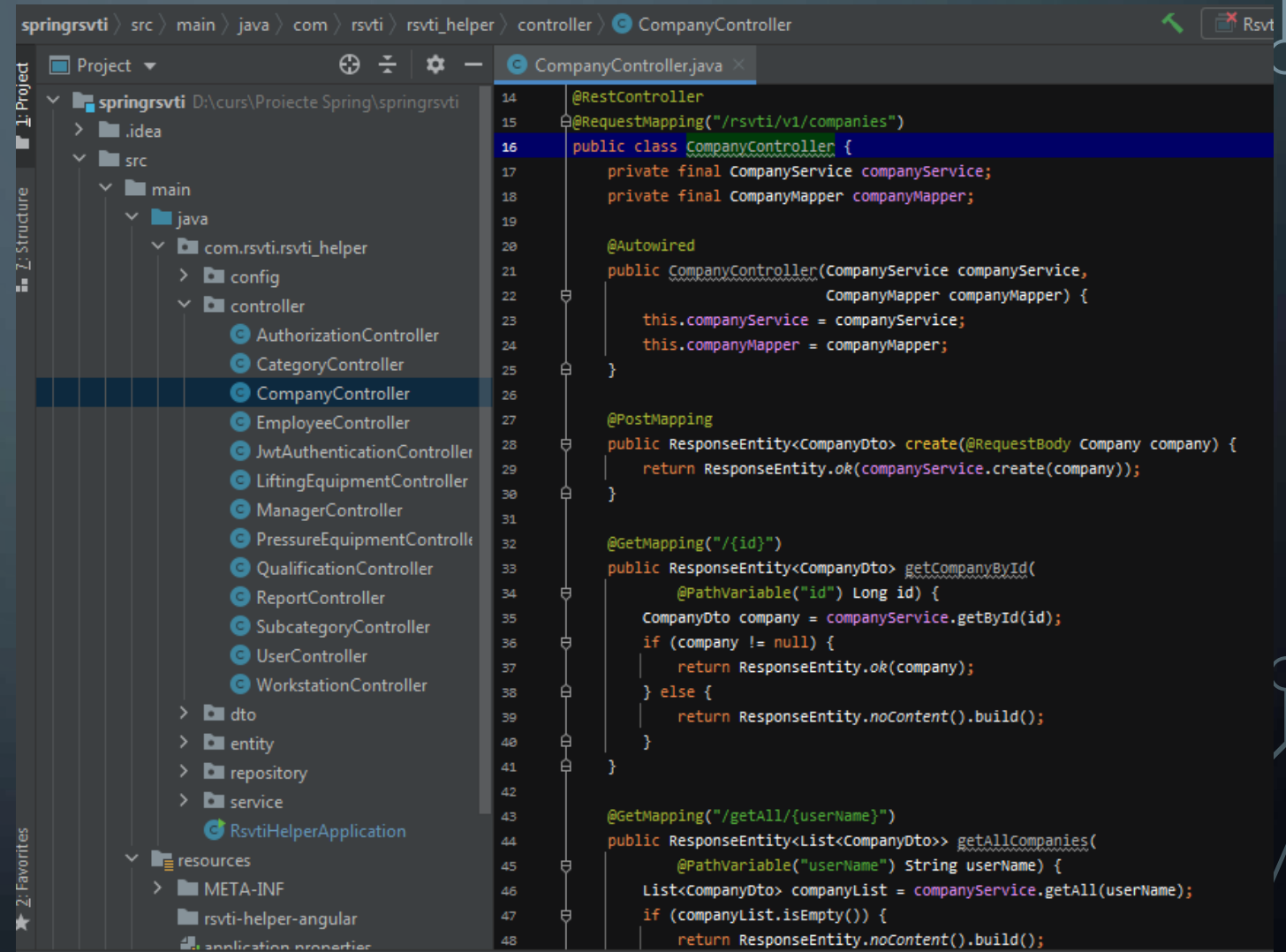
```
21 @EnableWebSecurity
22 @EnableGlobalMethodSecurity(prePostEnabled = true)
23 public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
24
25     private final JwtAuthenticationEntryPoint jwtAuthenticationEntryPoint;
26     private final UserDetailsService userDetailsService;
27     private final JwtRequestFilter jwtRequestFilter;
28
29     @Autowired
30     public WebSecurityConfig(
31         JwtAuthenticationEntryPoint jwtAuthenticationEntryPoint,
32         @Qualifier("DefaultUserDetailsServiceImpl") UserDetailsService userDetailsService,
33         JwtRequestFilter jwtRequestFilter
34     ) {
35         this.jwtAuthenticationEntryPoint = jwtAuthenticationEntryPoint;
36         this.userDetailsService = userDetailsService;
37         this.jwtRequestFilter = jwtRequestFilter;
38     }
39
40     @Override
41     protected void configure(HttpSecurity httpSecurity) throws Exception {
42         httpSecurity.cors().disable();
43         httpSecurity.csrf().disable().httpSecurity
44             .authorizeRequests().ExpressionUrlAuthorizationConfigurer<H>.ExpressionInterceptUrlRegistry
45             .antMatchers(...antPatterns: "/authenticate", "/rsvti/v1/users/register").permitAll().antMatchers
46             .permitAll().anyRequest().authenticated().ExpressionUrlAuthorizationConfigurer<H>.ExpressionInterceptUrlReg
47             .and().exceptionHandling().authenticationEntryPoint(jwtAuthenticationEntryPoint) ExceptionHandli
48             .and().sessionManagement().SessionManagementConfigurer<HttpSecurity>
49             .sessionCreationPolicy(SessionCreationPolicy.STATELESS);
50         httpSecurity.addFilterBefore(jwtRequestFilter, UsernamePasswordAuthenticationFilter.class);
51     }
52
53     @Bean
```



# Back-end : Spring

## Pachetul Controller

- Conține clasele adnotate cu `@RestController` care vor expune prin end-point-uri, pentru aplicația de front-end, metode prin care se pot executa operațiunile necesare de adăugare, editare, ștergere și primire de date
- Pe metode sunt folosite, după caz, adnotări pentru identificarea metodei HTTP folosite, `@GetMapping()`, `@PostMapping()`, `@PutMapping()`, `@DeleteMapping()`



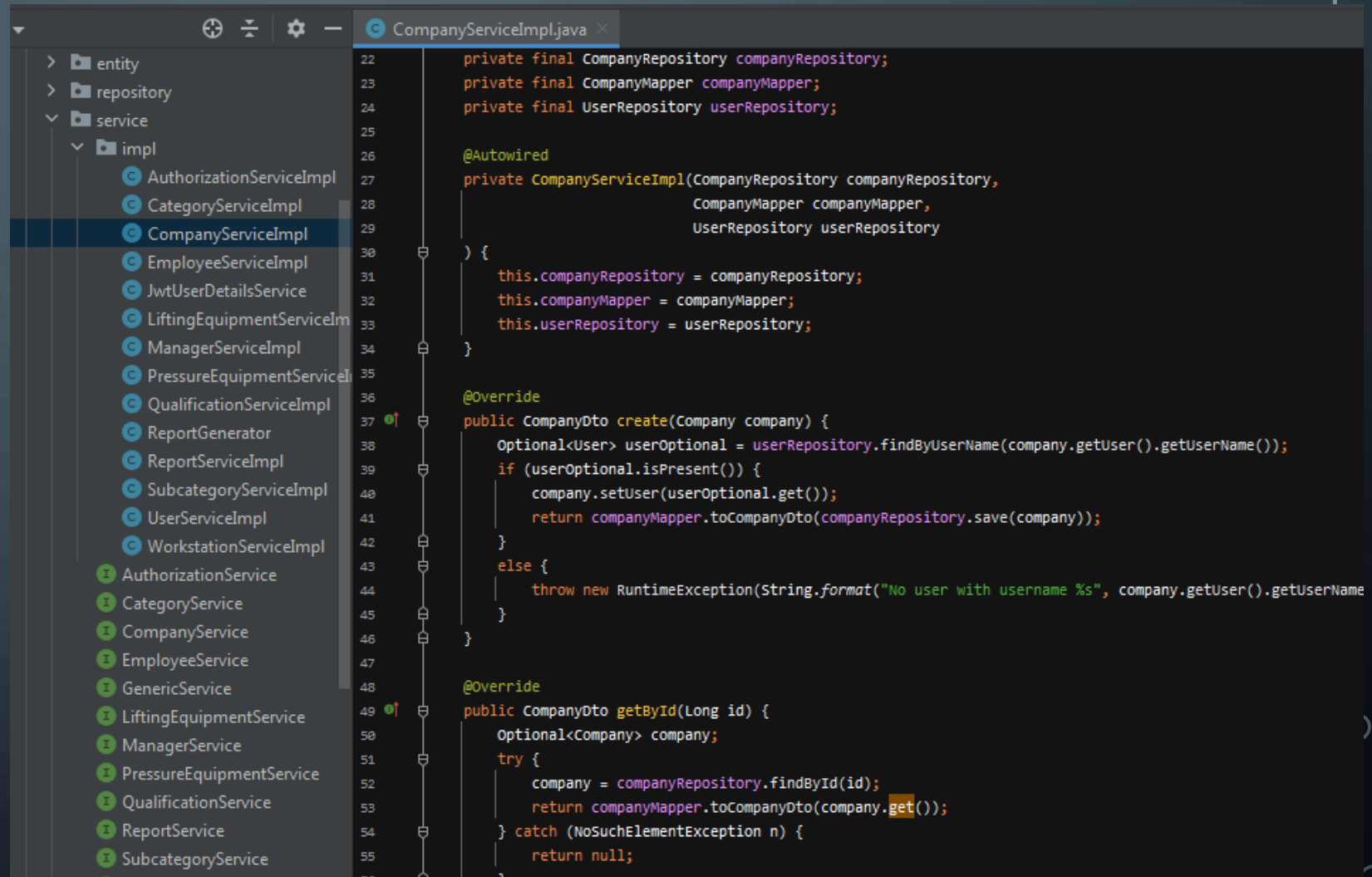
The screenshot displays an IDE interface with two main panels. The left panel shows the 'Project' structure of a Spring Boot application named 'springrsvti'. The 'src/main/java/com/rsvti/rsvti\_helper/controller' package is expanded, listing various controllers, with 'CompanyController' highlighted. The right panel shows the source code of 'CompanyController.java'. The code is annotated with Spring annotations and implements RESTful endpoints for company management.

```
14 @RestController
15 @RequestMapping("/rsvti/v1/companies")
16 public class CompanyController {
17     private final CompanyService companyService;
18     private final CompanyMapper companyMapper;
19
20     @Autowired
21     public CompanyController(CompanyService companyService,
22                             CompanyMapper companyMapper) {
23         this.companyService = companyService;
24         this.companyMapper = companyMapper;
25     }
26
27     @PostMapping
28     public ResponseEntity<CompanyDto> create(@RequestBody Company company) {
29         return ResponseEntity.ok(companyService.create(company));
30     }
31
32     @GetMapping("/{id}")
33     public ResponseEntity<CompanyDto> getCompanyById(
34         @PathVariable("id") Long id) {
35         CompanyDto company = companyService.getById(id);
36         if (company != null) {
37             return ResponseEntity.ok(company);
38         } else {
39             return ResponseEntity.noContent().build();
40         }
41     }
42
43     @GetMapping("/getAll/{userName}")
44     public ResponseEntity<List<CompanyDto>> getAllCompanies(
45         @PathVariable("userName") String userName) {
46         List<CompanyDto> companyList = companyService.getAll(userName);
47         if (companyList.isEmpty()) {
48             return ResponseEntity.noContent().build();
49         }
50     }
51 }
```

# Back-end : Spring

## Pachetul Service

- Conține clasele adnotate cu `@Service` cu logica de business a aplicației
- Preia obiecte și le transformă prin clasele Mapper din DTO (Data Transfer Object) în Entity și invers, după caz



The screenshot displays an IDE with a project structure on the left and the source code of `CompanyServiceImpl.java` on the right.

**Project Structure (Left Panel):**

- `entity`
- `repository`
- `service`
  - `impl`
    - `AuthorizationServiceImpl`
    - `CategoryServiceImpl`
    - `CompanyServiceImpl`**
    - `EmployeeServiceImpl`
    - `JwtUserDetailsService`
    - `LiftingEquipmentServiceImpl`
    - `ManagerServiceImpl`
    - `PressureEquipmentServiceImpl`
    - `QualificationServiceImpl`
    - `ReportGenerator`
    - `ReportServiceImpl`
    - `SubcategoryServiceImpl`
    - `UserServiceImpl`
    - `WorkstationServiceImpl`
  - `AuthorizationService`
  - `CategoryService`
  - `CompanyService`
  - `EmployeeService`
  - `GenericService`
  - `LiftingEquipmentService`
  - `ManagerService`
  - `PressureEquipmentService`
  - `QualificationService`
  - `ReportService`
  - `SubcategoryService`

**Source Code (Right Panel):**

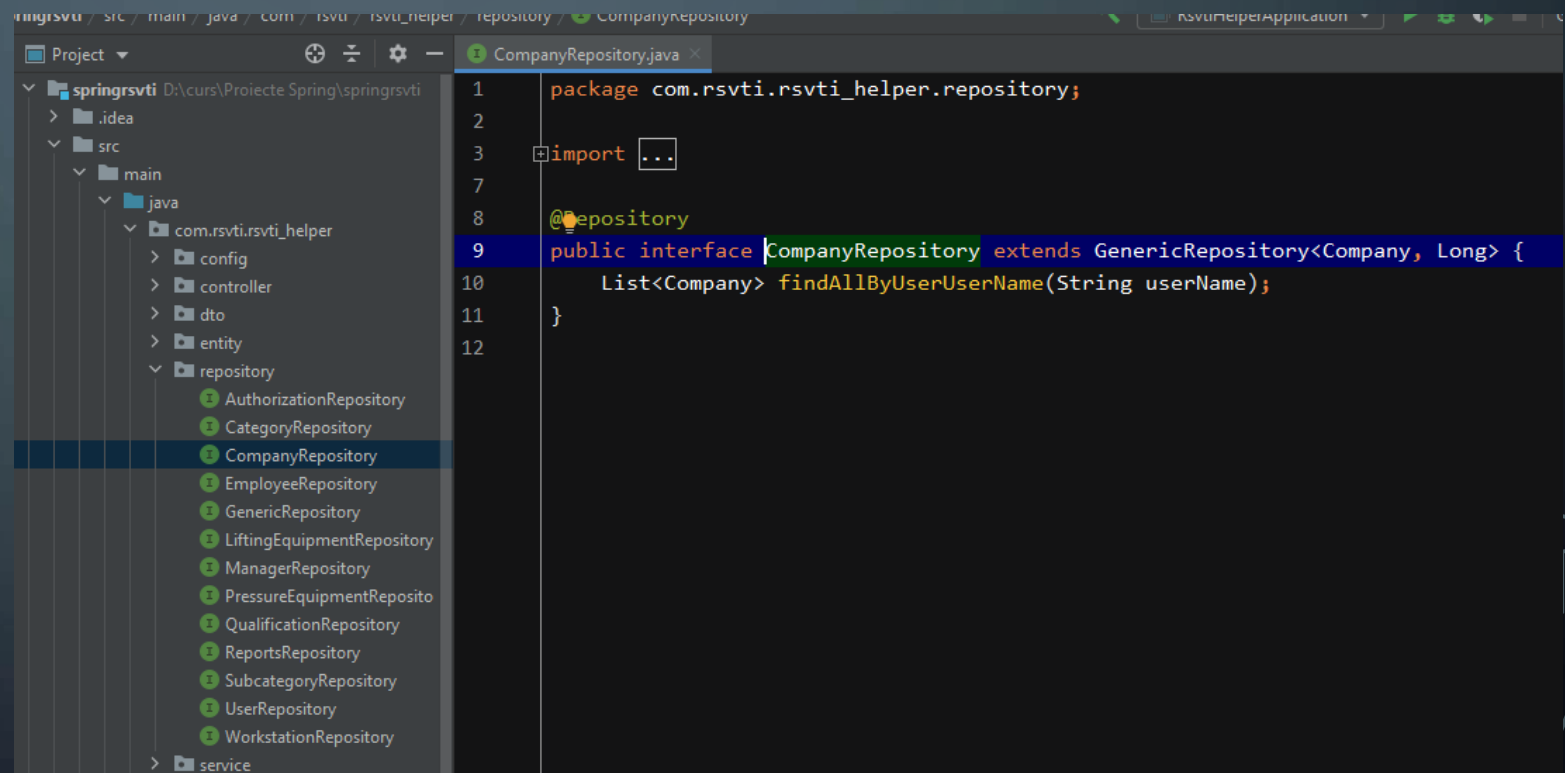
```
22 private final CompanyRepository companyRepository;  
23 private final CompanyMapper companyMapper;  
24 private final UserRepository userRepository;  
25  
26 @Autowired  
27 private CompanyServiceImpl(CompanyRepository companyRepository,  
28                             CompanyMapper companyMapper,  
29                             UserRepository userRepository)  
30 {  
31     this.companyRepository = companyRepository;  
32     this.companyMapper = companyMapper;  
33     this.userRepository = userRepository;  
34 }  
35  
36 @Override  
37 public CompanyDto create(Company company) {  
38     Optional<User> userOptional = userRepository.findByUserName(company.getUser().getUserName());  
39     if (userOptional.isPresent()) {  
40         company.setUser(userOptional.get());  
41         return companyMapper.toCompanyDto(companyRepository.save(company));  
42     }  
43     else {  
44         throw new RuntimeException(String.format("No user with username %s", company.getUser().getUserName()));  
45     }  
46 }  
47  
48 @Override  
49 public CompanyDto getById(Long id) {  
50     Optional<Company> company;  
51     try {  
52         company = companyRepository.findById(id);  
53         return companyMapper.toCompanyDto(company.get());  
54     } catch (NoSuchElementException n) {  
55         return null;  
56     }
```

# Back-end : Spring

**Pachetul Repository** conține interfețele Repository adnotate cu `@Repository` care fac posibilă manipularea datelor introduse în baza de date cu ajutorul unor metode predefinite

În spate, Repository folosește JDBC (Java DataBase Connector) cu ajutorul căruia poate executa query-uri în MySQL.

În interfața Repository, dacă nu este disponibilă o metodă predefinită potrivită cu ceea ce se dorește, prin numele dat metodei, va putea crea un query.



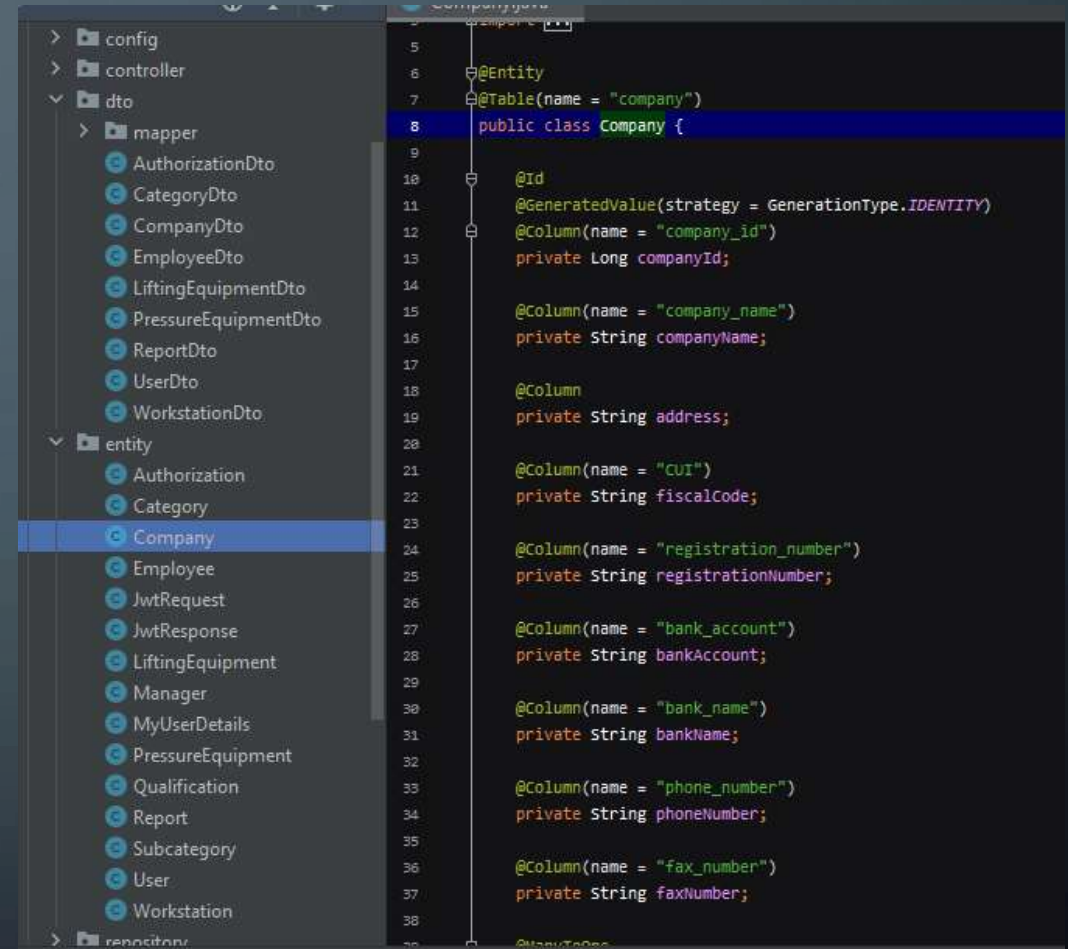
# Back-end : Spring

**Pachetul Entity** conține clasele care descriu obiectele folosite de aplicație și care trebuie să fie potrivite cu tabelele din baza de date.

Numele clasei va fi potrivit cu numele tabelului, iar variabilele de instanță se vor potrivi cu numele coloanelor din tabel, astfel că un obiect va fi mapat la un rând din tabel.

**Pachetul Dto** conține clasele care descriu obiectele care vor fi trimise prin Controller, astfel că aplicația front-end nu va avea acces direct la informațiile din baza de date.

**Pachetul Mapper** conține clasele care vor transforma entitățile în DTO și invers



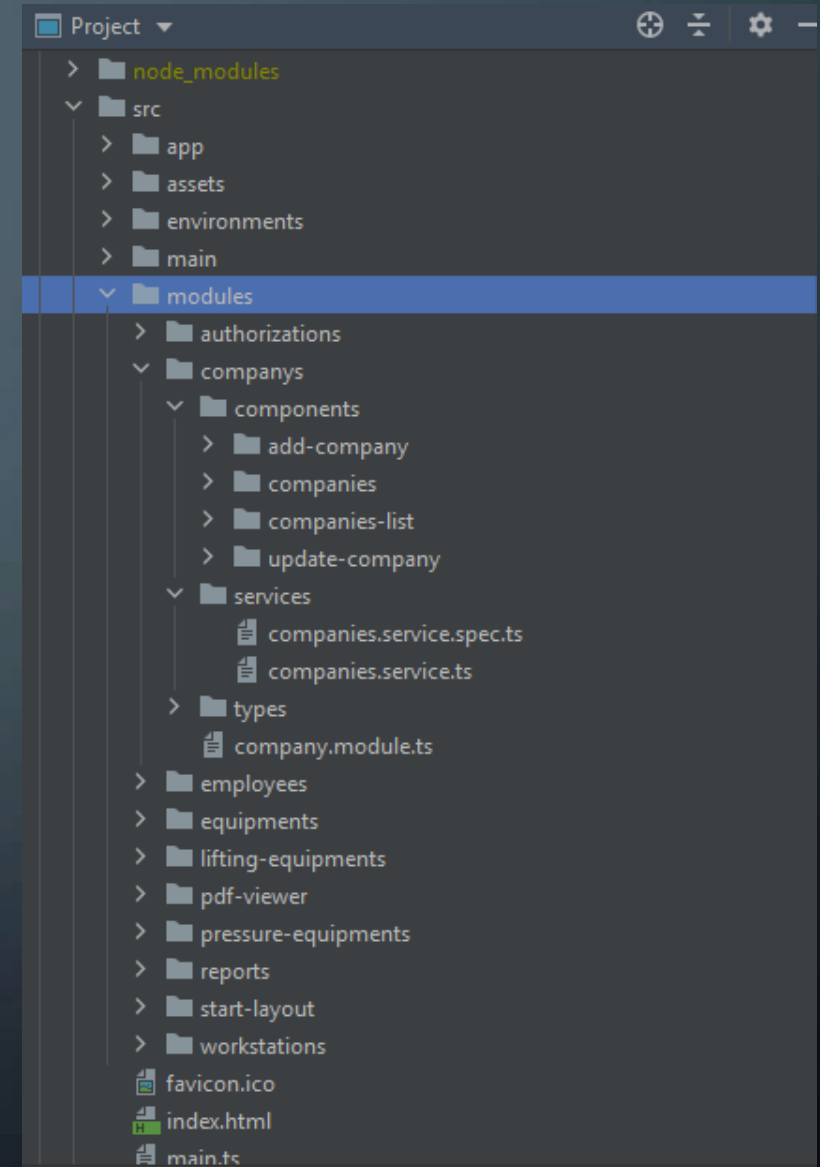
The screenshot displays an IDE with a project structure on the left and the source code of the `Company` entity on the right. The project structure shows a package hierarchy: `config`, `controller`, `dto` (containing `mapper`), and `entity`. The `entity` package is expanded, showing various entity classes, with `Company` selected. The `Company` class is annotated with `@Entity` and `@Table(name = "company")`. It is a public class with the following attributes: `@Id` `@GeneratedValue(strategy = GenerationType.IDENTITY)` `@Column(name = "company_id")` `private Long companyId;` `@Column(name = "company_name")` `private String companyName;` `@Column` `private String address;` `@Column(name = "cui")` `private String fiscalCode;` `@Column(name = "registration_number")` `private String registrationNumber;` `@Column(name = "bank_account")` `private String bankAccount;` `@Column(name = "bank_name")` `private String bankName;` `@Column(name = "phone_number")` `private String phoneNumber;` `@Column(name = "fax_number")` `private String faxNumber;`

# Front-end : Angular

Partea de front-end este dezvoltată cu tehnologia Angular și este astfel compusă din module, fiecare modul pentru un obiect.

Fiecare modul conține:

1. Components - câte una pentru fiecare comportament
2. Service – pentru logica de business a aplicației
3. Type – modelul obiectelor după care vor fi mapate obiectele JSON primite

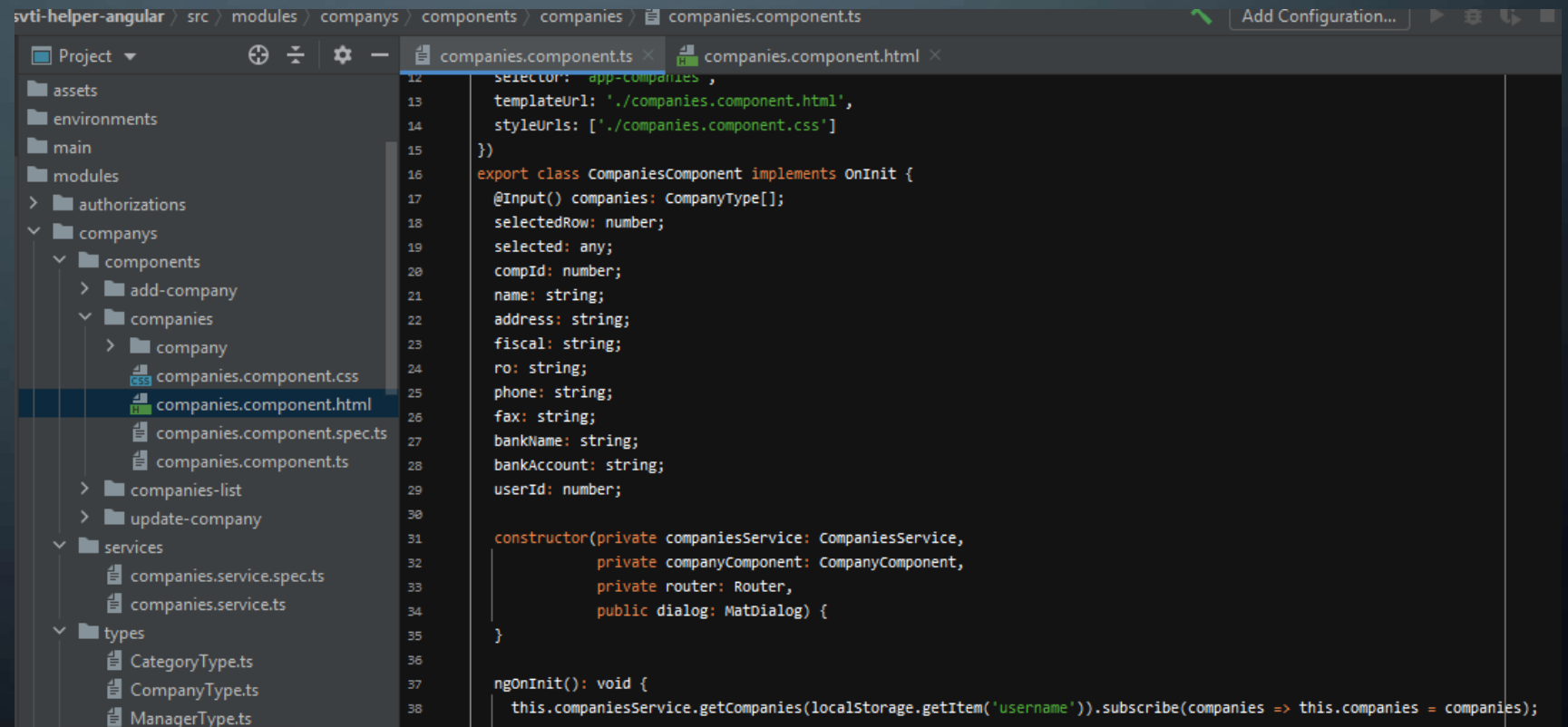




# Front-end : Angular

Ca și componente, într-un modul sunt componente pentru prezentarea listelor de obiecte, a obiectelor individuale, adăugare, editare etc.

O componentă este formată dintr-un fișier `.ts` care aduce obiectul pentru prezentare și un fișier `.html` care prezintă vizual obiectul din fișierul TypeScript. Mai conține și un fișier `.css` pentru stilizarea părții vizuale a componentei.



The screenshot shows an IDE with a project explorer on the left and a code editor on the right. The project explorer displays the following structure:

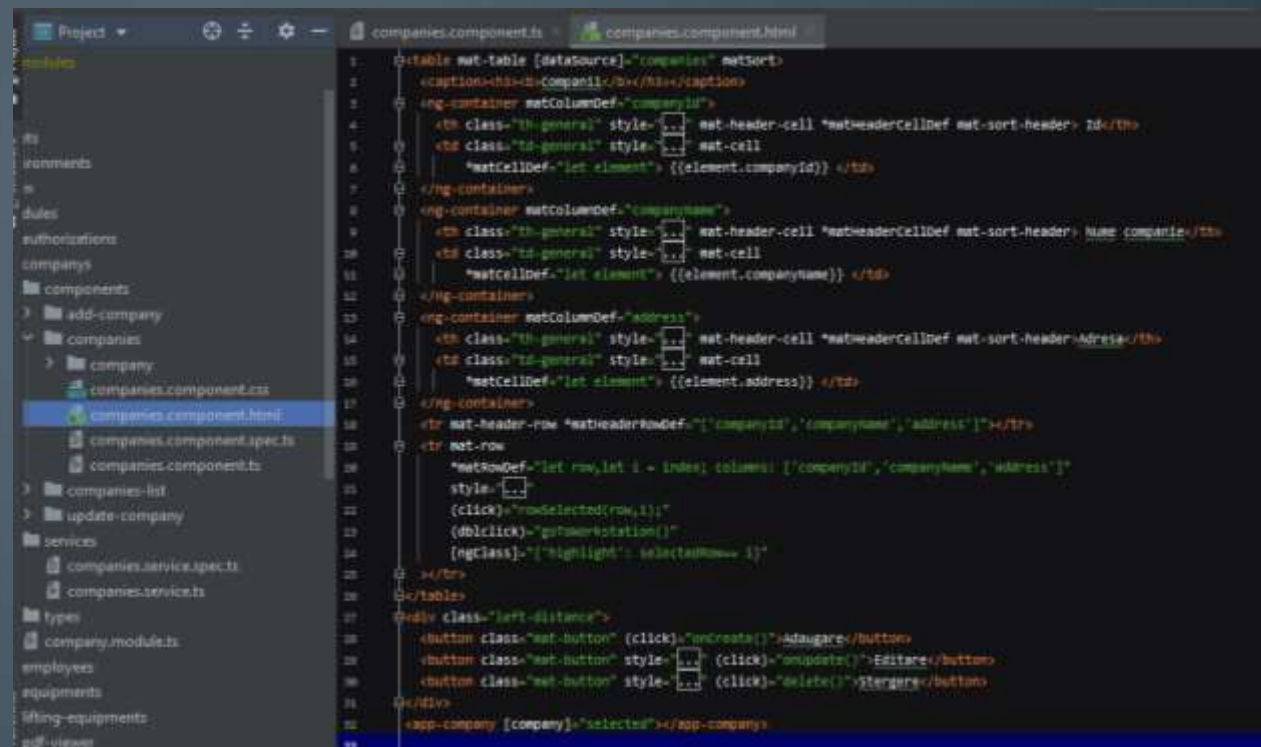
- assets
- environments
- main
- modules
  - authorizations
  - companys
    - components
      - add-company
      - companies
        - company
        - companies.component.css
        - companies.component.html
        - companies.component.spec.ts
        - companies.component.ts
      - companies-list
      - update-company
    - services
      - companies.service.spec.ts
      - companies.service.ts
    - types
      - CategoryType.ts
      - CompanyType.ts
      - ManagerType.ts

The code editor shows the `companies.component.ts` file with the following code:

```
12 selector: 'app-companies',
13 templateUrl: './companies.component.html',
14 styleUrls: ['./companies.component.css']
15 })
16 export class CompaniesComponent implements OnInit {
17   @Input() companies: CompanyType[];
18   selectedRow: number;
19   selected: any;
20   compId: number;
21   name: string;
22   address: string;
23   fiscal: string;
24   ro: string;
25   phone: string;
26   fax: string;
27   bankName: string;
28   bankAccount: string;
29   userId: number;
30
31   constructor(private companiesService: CompaniesService,
32               private companyComponent: CompanyComponent,
33               private router: Router,
34               public dialog: MatDialog) {
35   }
36
37   ngOnInit(): void {
38     this.companiesService.getCompanies(localStorage.getItem('username')).subscribe(companies => this.companies = companies);
39   }
40 }
```

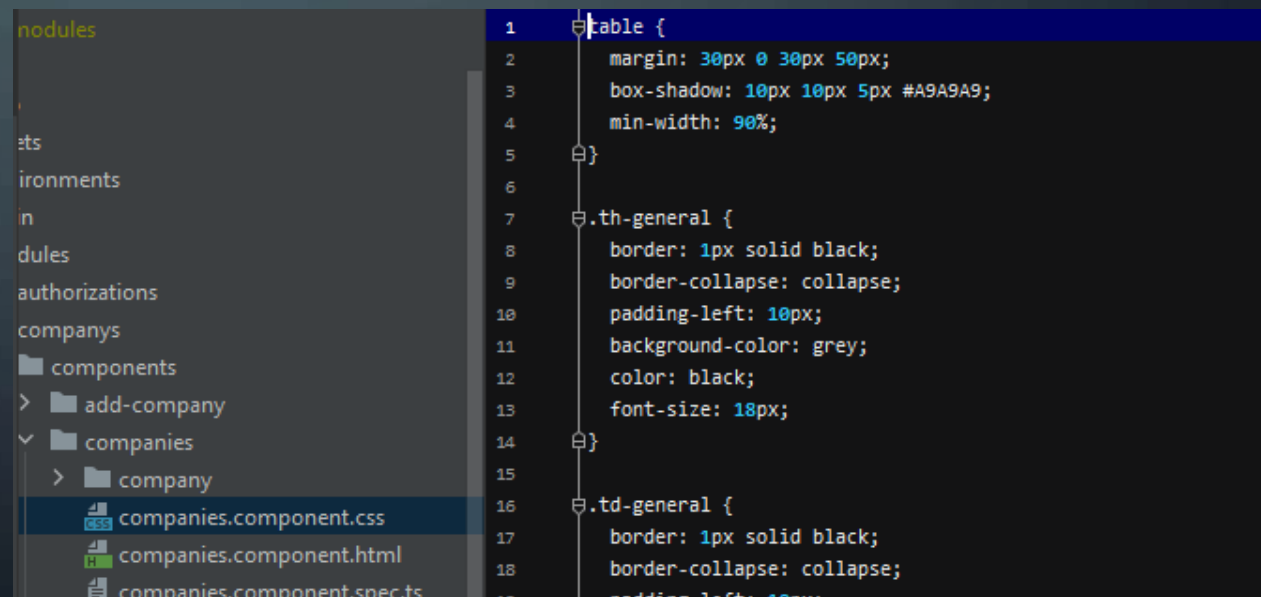
# Front-end : Angular

Fisier .html de prezentare vizuală  
-pagina web-



```
1 <table mat-table [dataSource]="companies" matSort>
2   <caption>this.company</caption>
3   <ng-container matColumnDef="companyId">
4     <th class="th-general" style="width: 33.33333333333333%; text-align: left;">ID</th>
5     <td class="td-general" style="width: 33.33333333333333%; text-align: left;">
6       <mat-cell-def "let element"> {{element.companyId}} </mat-cell-def>
7     </td>
8   </ng-container>
9   <ng-container matColumnDef="companyName">
10    <th class="th-general" style="width: 33.33333333333333%; text-align: left;">Nume companie</th>
11    <td class="td-general" style="width: 33.33333333333333%; text-align: left;">
12      <mat-cell-def "let element"> {{element.companyName}} </mat-cell-def>
13    </td>
14   </ng-container>
15   <ng-container matColumnDef="address">
16    <th class="th-general" style="width: 33.33333333333333%; text-align: left;">Adresa</th>
17    <td class="td-general" style="width: 33.33333333333333%; text-align: left;">
18      <mat-cell-def "let element"> {{element.address}} </mat-cell-def>
19    </td>
20   </ng-container>
21   <tr mat-header-row "matHeaderRowDef"=["companyId","companyName","address"]></tr>
22   <tr mat-row "matRowDef"=["let row,let i = index; columns: ['companyId','companyName','address']"]>
23     <td class="td-general" style="width: 33.33333333333333%; text-align: left;">
24       <mat-cell-def "let element"> {{element.address}} </mat-cell-def>
25     </td>
26   </tr>
27 </table>
28 <div class="left-distances">
29   <button class="mat-button" (click)="onCreate()">Adauga</button>
30   <button class="mat-button" style="margin-left: 10px;" (click)="onUpdate()">Editare</button>
31   <button class="mat-button" style="margin-left: 10px;" (click)="onDelete()">Stergere</button>
32 </div>
33 <app-company [company]="selected"></app-company>
```

Fisier .css de stilizare a paginii web



```
1 table {
2   margin: 30px 0 30px 50px;
3   box-shadow: 10px 10px 5px #A9A9A9;
4   min-width: 90%;
5 }
6
7 .th-general {
8   border: 1px solid black;
9   border-collapse: collapse;
10  padding-left: 10px;
11  background-color: grey;
12  color: black;
13  font-size: 18px;
14 }
15
16 .td-general {
17   border: 1px solid black;
18   border-collapse: collapse;
19   padding-left: 10px;
```

# Front-end : Angular

Service este un fișier .ts care conține logica componentei și prin care sunt apelate și metodele HTTP accesând endpoint-urile expuse de back-end

```
54 getCompanies(value: string): Observable<CompanyType[]> {
55     return this.http.get<CompanyType[]>(this.url + '/getAll/' + value);
56 }
57
58 getCompanyById(id: number): Observable<CompanyType> {
59     return this.http.get<CompanyType>(this.url + '/getById/' + id);
60 }
61
62 createManager(company): Promise<any> {
63     return this.http.post<any>('http://localhost:8090/rsvt/v1/managers/create', {
64         firstName: company.managerFirstName,
65         lastName: company.managerLastName,
66         managerRole: company.managerRole
67     }).toPromise();
68 }
69
70 async addCompany(company): Promise<void> {
71     let createManager = await this.createManager(company);
72     return this.http.post<any>(this.url, {
73         companyName: company.companyName,
74         address: company.companyAddress,
75         fiscalCode: company.fiscal,
76         registrationNumber: company.ro,
77         bankName: company.bankName,
78         bankAccount: company.bankAccount,
79         phoneNuber: company.phone,
80         faxNumber: company.fax,
81         manager: createManager,
82         user: { 'username': localStorage.getItem('username') }
83     }).toPromise();
84 }
```

```
8 @Injectable({
9     providedIn: 'root'
10 })
11 export class CompaniesService {
12     url: string = 'http://localhost:8090/rsvt/v1/companies';
13     public selectedCompany: any;
14
15     constructor(private http: HttpClient) {
16     }
17
18     setData(selected): void {
19         this.selectedCompany = selected;
20     }
21
22     getData(): any {
23         return this.selectedCompany;
24     }
25
26     form: FormGroup = new FormGroup({
27         key: new FormControl(''),
28         companyName: new FormControl('', Validators.required),
29         companyAddress: new FormControl('', Validators.required),
30         fiscal: new FormControl('', Validators.required),
31         ro: new FormControl('', Validators.required),
32         phone: new FormControl(''),
33         fax: new FormControl(''),
34         bankName: new FormControl('', Validators.required),
35         bankAccount: new FormControl('', Validators.required),
36         managerFirstName: new FormControl('', Validators.required),
37         managerLastName: new FormControl('', Validators.required),
38         managerRole: new FormControl('', Validators.required)
39     });
40 }
```

```
getUserByUserName(username: string){
    return this.http.get<any>('http://localhost:8090/rsvt/v1/users/getByUserName/' + username);
}

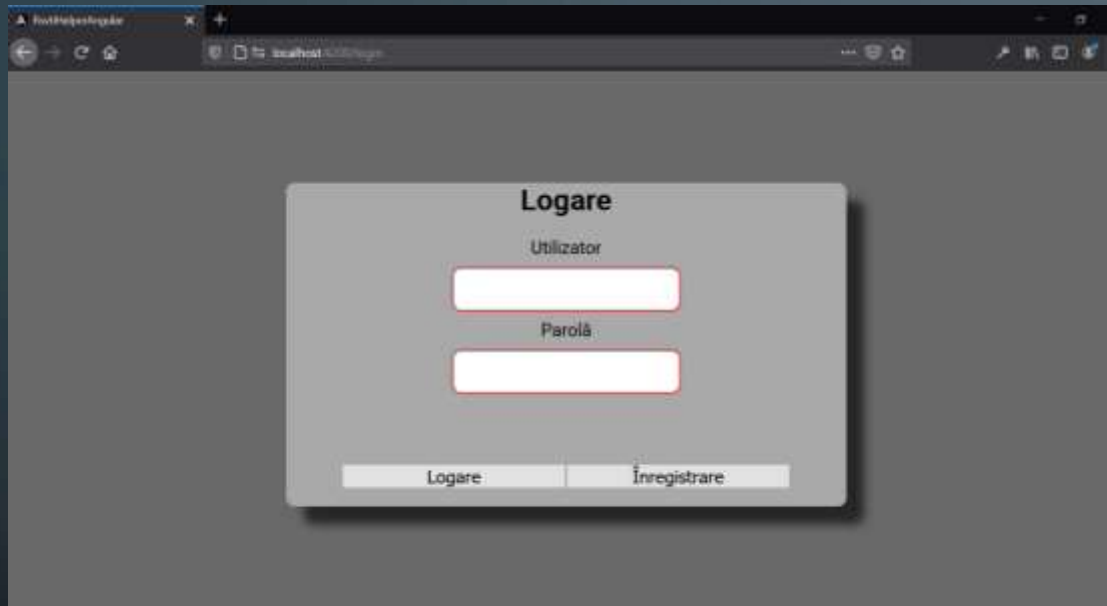
getManagerById(id: number): Observable<ManagerType> {
    return this.http.get<any>('http://localhost:8090/rsvt/v1/managers/getManagerById/' + id);
}

updateCompany(company): Promise<void> {
    return this.http.put<any>(this.url, {
        companyId: company.companyId,
        companyName: company.companyName,
        address: company.companyAddress,
        fiscalCode: company.fiscal,
        registrationNumber: company.ro,
        bankName: company.bankName,
        bankAccount: company.bankAccount,
        phoneNuber: company.phone,
        faxNumber: company.fax,
        managerId: company.managerId,
        userName: localStorage.getItem('username')
    }).toPromise();
}

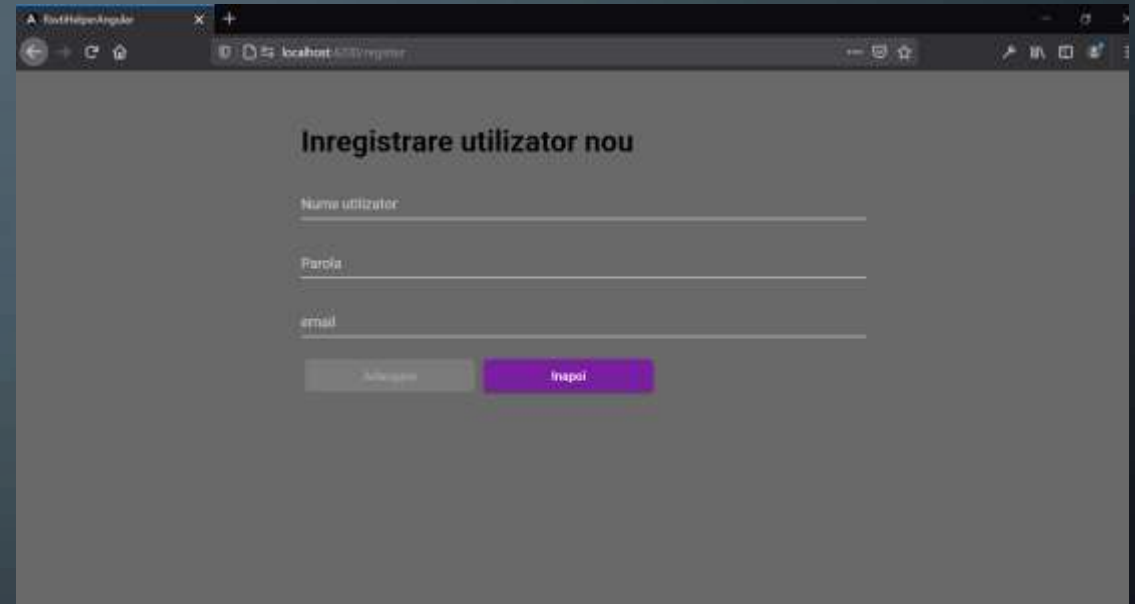
deleteCompany(id: number): Promise<any> {
    return this.http.delete(this.url + '/' + id).toPromise();
}
```

# Rularea

## Login / înregistrare utilizator nou



Aplicația nu poate fi accesată fără a fi logat un utilizator. Logarea duce la selectarea companiilor pe care utilizatorul logat le are sub supraveghere.



Dacă utilizatorul nu este înregistrat poate să o facă cu ajutorul butonului **Înregistrare**, datele acestuia fiind salvate în baza de date.

## PAGINA PRINCIPALĂ

### MENIU

### COMPANII

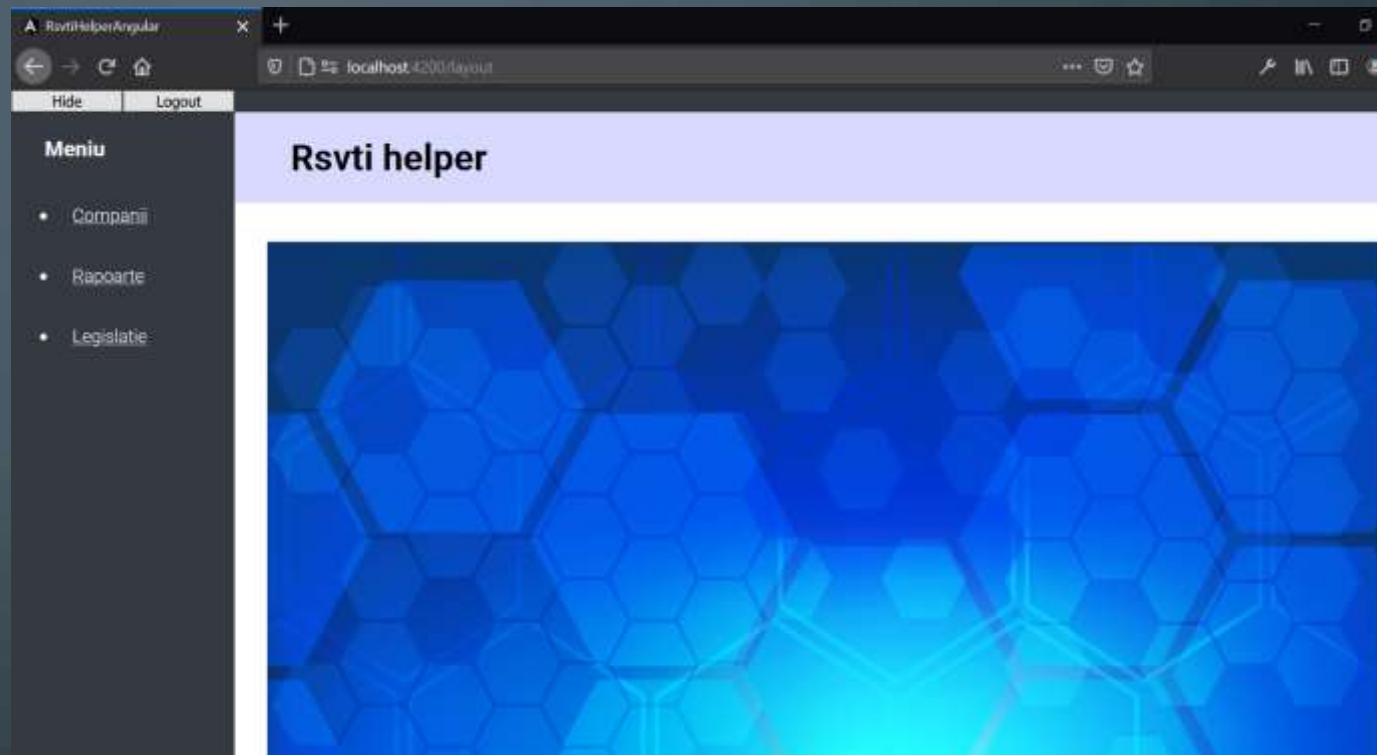
- Vizualizează companiile cu toți angajații și instalațiile acestora

### RAPOARTE

- Scoate diferite rapoarte

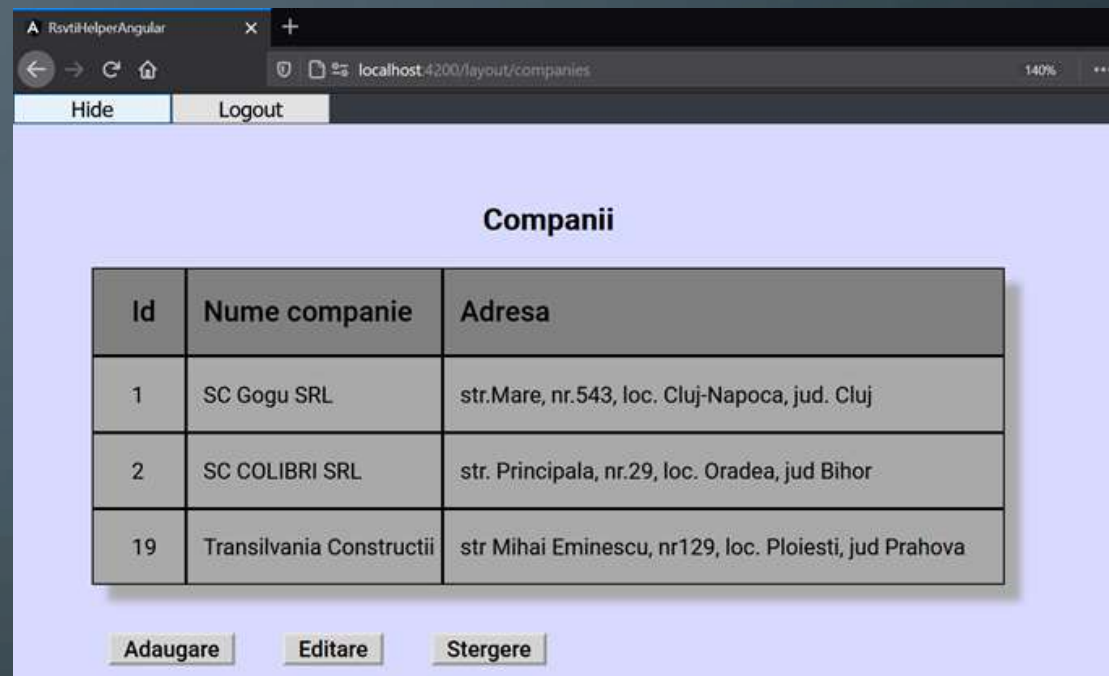
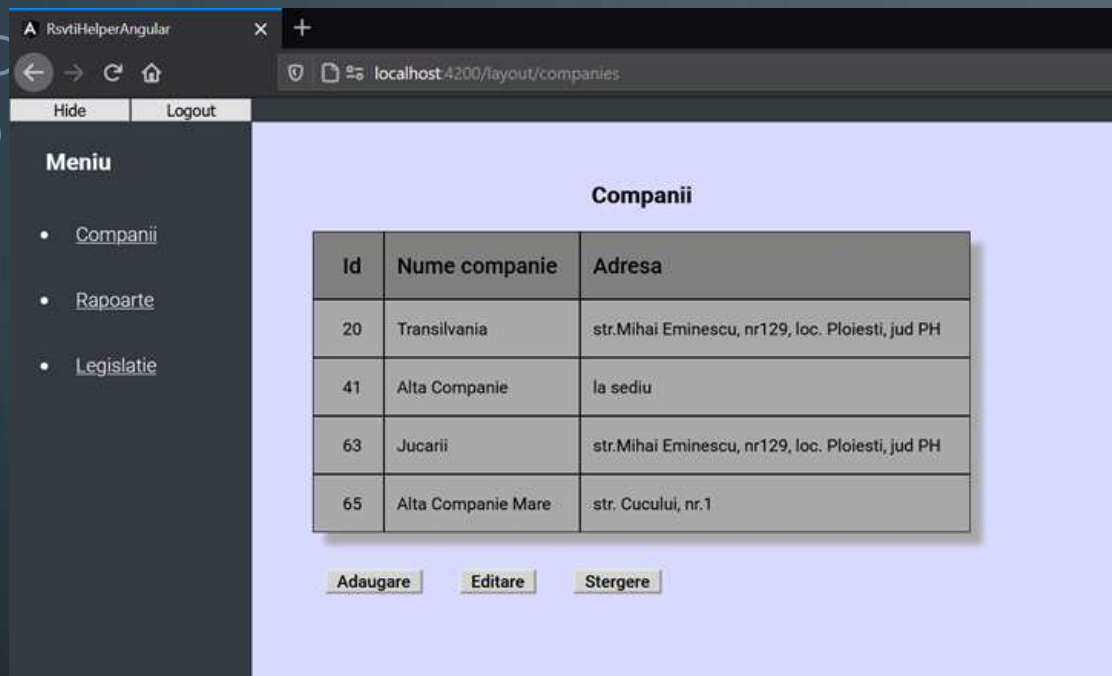
### LEGISLAȚIE

- Deschide documente care conțin legile după care își desfășoară activitatea companiile





# Afisarea companiilor



Companii afișate în funcție de utilizatorul logat

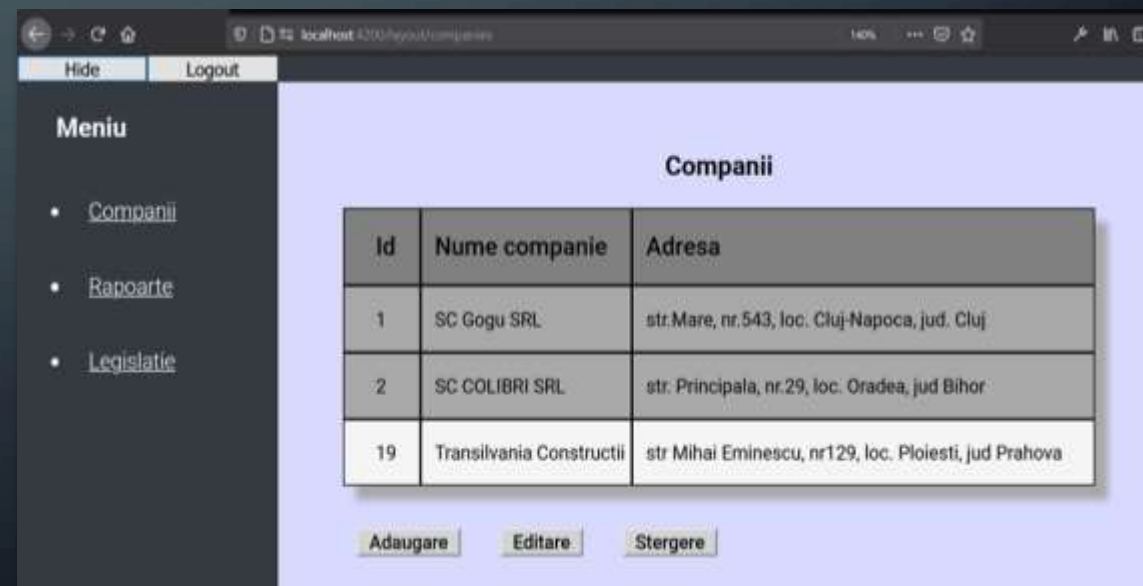
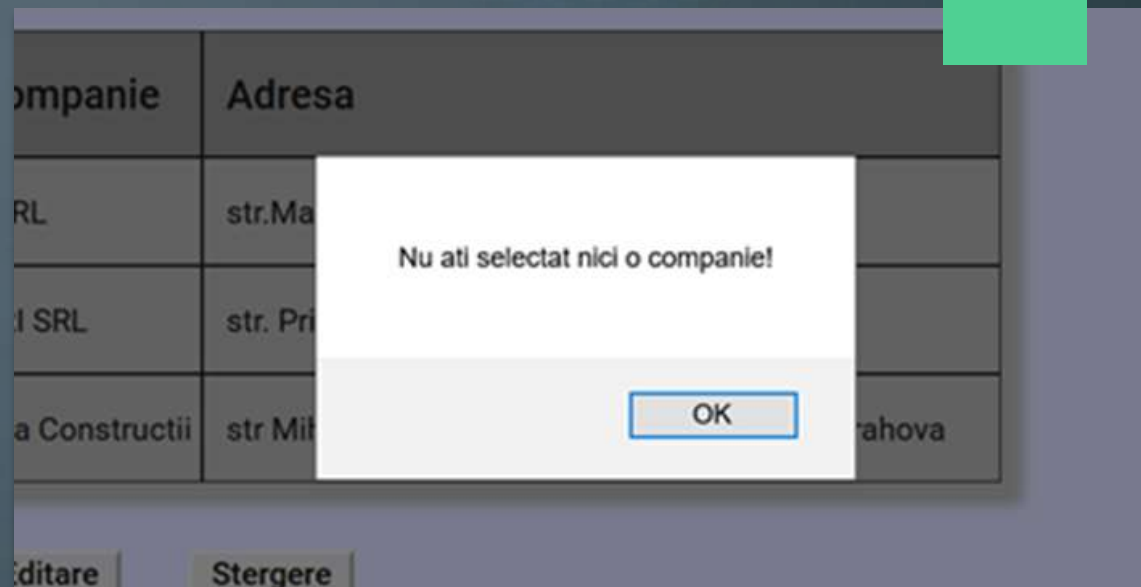
- **Buton Hide** - Meniul lateral poate fi ascuns pentru o vizualizare mai bună a datelor
- **Buton Logout** - Utilizatorul se poate deloga

# Companii

Comaniile sunt afișate într-un tabel pe rânduri selectabile

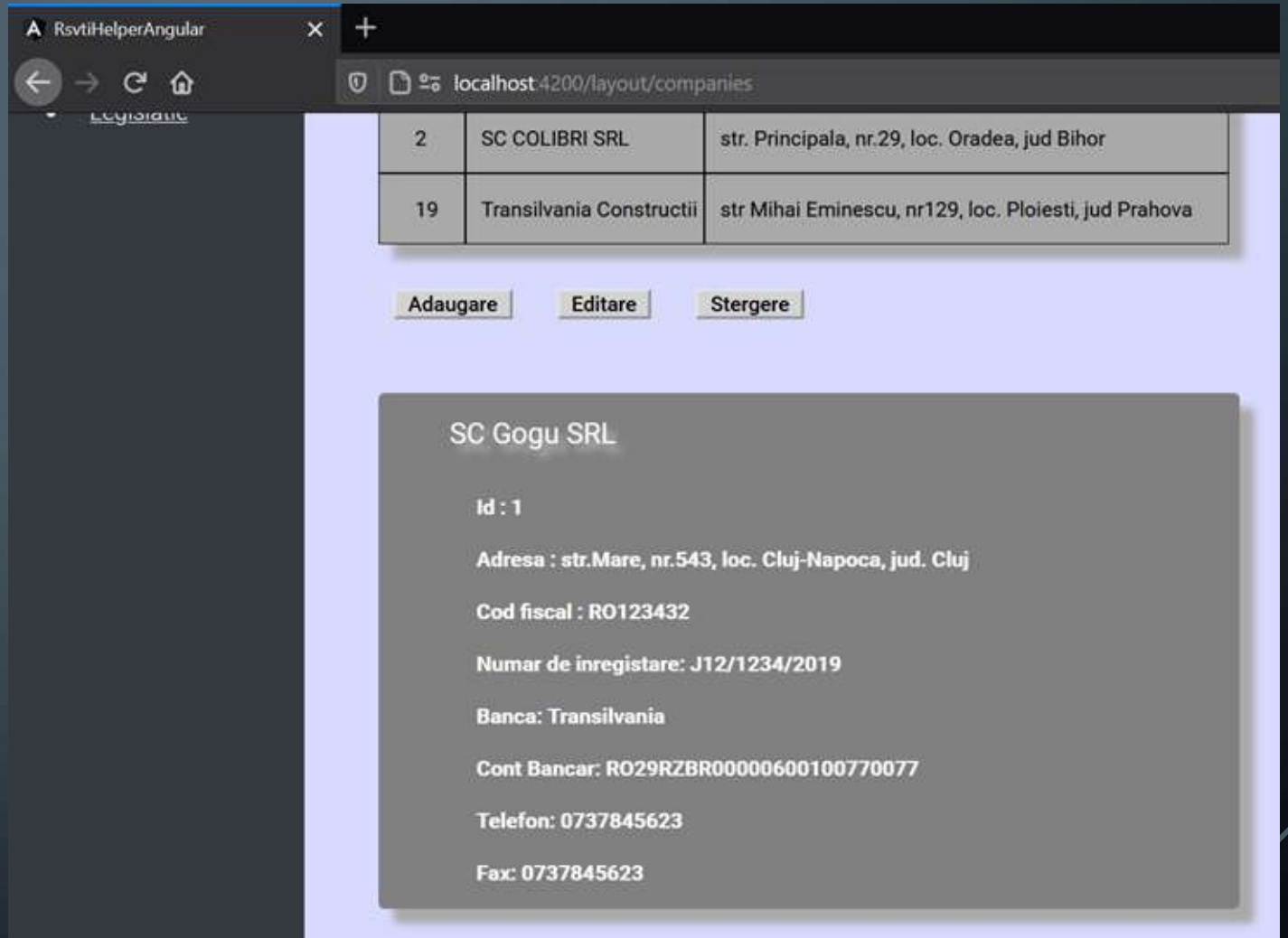
## Butoane:

- **Adăugare** - utilizatorul adaugă o companie nouă în baza de date
- **Editare** – utilizatorul editează una dintre companiile existente
- **Stergere** – utilizatorul șterge o companie existentă din baza de date



## Detalii companie

Cu o selectare simplă, în partea de jos a paginii se vor afișa informațiile detaliate ale companiei selectate din tabel.



The screenshot shows a web browser window with the title 'RsvtiHelperAngular'. The address bar displays 'localhost:4200/layout/companies'. The page content includes a table with two rows of company data. Below the table are three buttons: 'Adaugare', 'Editare', and 'Stergere'. A modal window is open, displaying detailed information for 'SC Gogu SRL'.

2	SC COLIBRI SRL	str. Principala, nr.29, loc. Oradea, jud Bihor
19	Transilvania Constructii	str Mihai Eminescu, nr129, loc. Ploiesti, jud Prahova

[Adaugare](#) [Editare](#) [Stergere](#)

**SC Gogu SRL**

**Id :** 1

**Adresa :** str.Mare, nr.543, loc. Cluj-Napoca, jud. Cluj

**Cod fiscal :** RO123432

**Numar de inregistrare:** J12/1234/2019

**Banca:** Transilvania

**Cont Bancar:** RO29RZBR00000600100770077

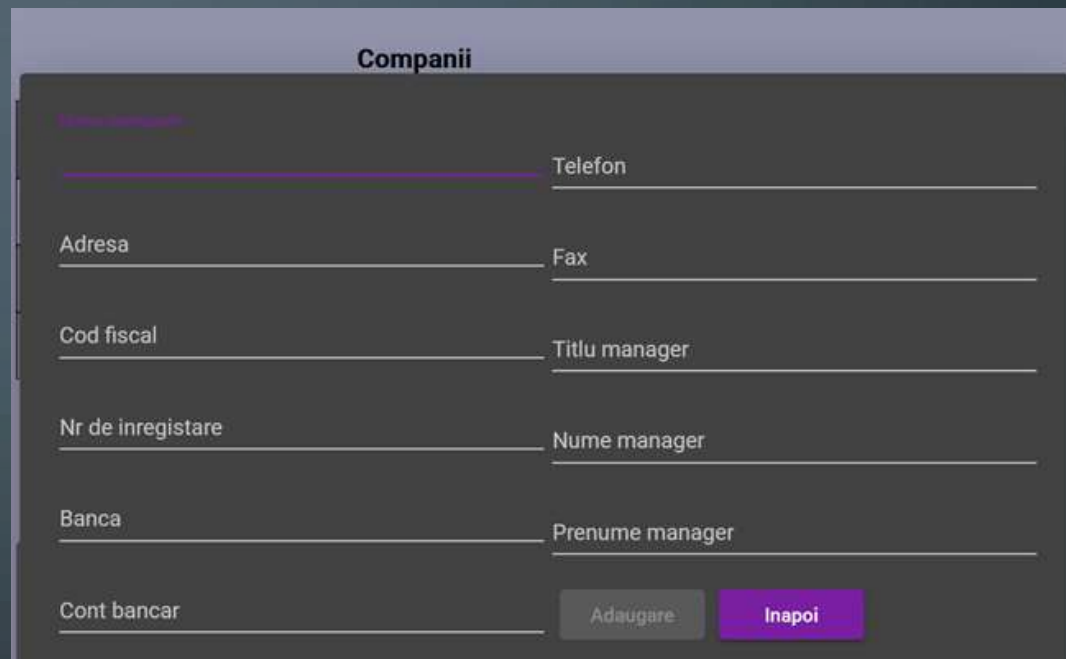
**Telefon:** 0737845623

**Fax:** 0737845623

# Adăugarea unei companii

Se face într-o fereastră tip dialog în care avem:

- Câmpuri de introducere a datelor
- Butonul **Adaugă** – adaugă în baza de date o companie nouă cu datele colectate din **input**-uri. *Butonul nu este activ dacă câmpurile obligatorii nu sunt completate.*
- Butonul **Înapoi** – închide fereastra de tip dialog



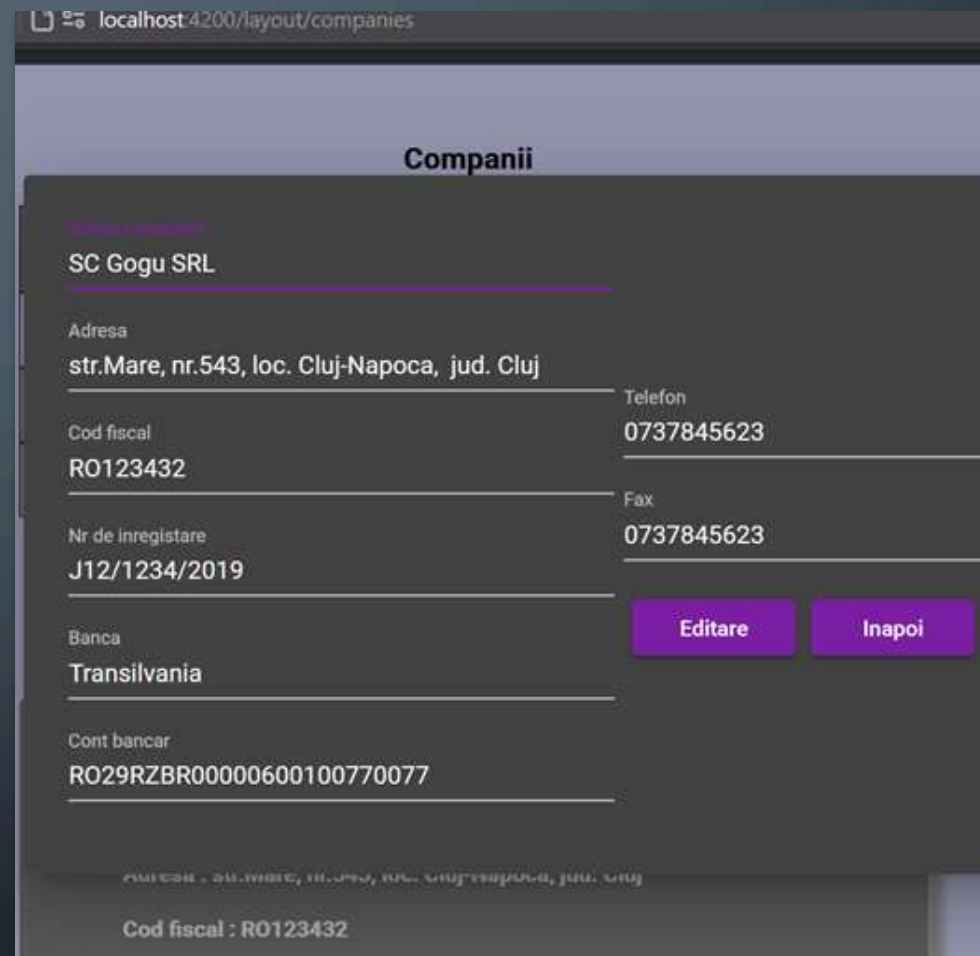
The screenshot shows a dialog box titled "Companii" with a light blue header. The main area is white and contains several input fields arranged in two columns. The fields are: "Nume companie" (highlighted in red), "Telefon", "Adresa", "Fax", "Cod fiscal", "Titlu manager", "Nr de inregistrare", "Nume manager", "Banca", "Prenume manager", and "Cont bancar". At the bottom right, there are two buttons: "Adaugare" (disabled, grey) and "Inapoi" (active, red).

Companii	
Nume companie	
	Telefon
Adresa	Fax
Cod fiscal	Titlu manager
Nr de inregistrare	Nume manager
Banca	Prenume manager
Cont bancar	Adaugare Inapoi

# Editarea unei companii existente

Se face într-o fereastră tip dialog în care avem:

- Câmpuri de introducere a datelor completate automat cu datele actuale ale companiei.
- Butonul **Adaugă** – salvează în baza de date noile date ale companiei selectate.
- Butonul **Înapoi** – închide fereastra de tip dialog



The screenshot shows a web browser window with the address bar displaying 'localhost:4200/layout/companies'. The page title is 'Companii'. The form contains the following fields and values:

SC Gogu SRL	
Adresa	str.Mare, nr.543, loc. Cluj-Napoca, jud. Cluj
Cod fiscal	RO123432
Telefon	0737845623
Nr de inregistrare	J12/1234/2019
Fax	0737845623
Banca	Transilvania
Cont bancar	RO29RZBR00000600100770077

At the bottom of the form, there are two buttons: 'Editare' and 'Inapoi'. Below the form, there is a summary line: 'Adresa : str.Mare, nr.543, loc. Cluj-Napoca, jud. Cluj' and 'Cod fiscal : RO123432'.



# Puncte de lucru

Cu dublu click pe o companie afișăm pagina cu punctele de lucru ale respectivei companii, care conține:

➤ Un tabel cu toate punctele de lucru ale companiei selectate

➤ **Butoane:**

**Adăugare** – adaugă un nou punct de lucru în baza de date

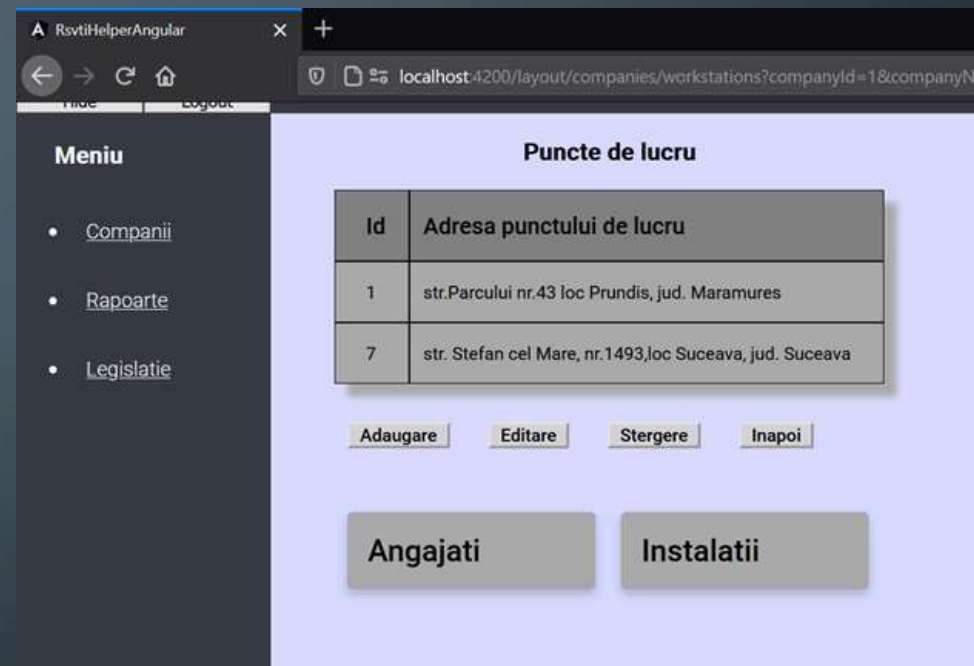
**Editare** – editează datele unui punct de lucru existent

**Ștergere** – șterge un punct de lucru din baza de date

**Înapoi** – afișează din nou pagina **Companii**

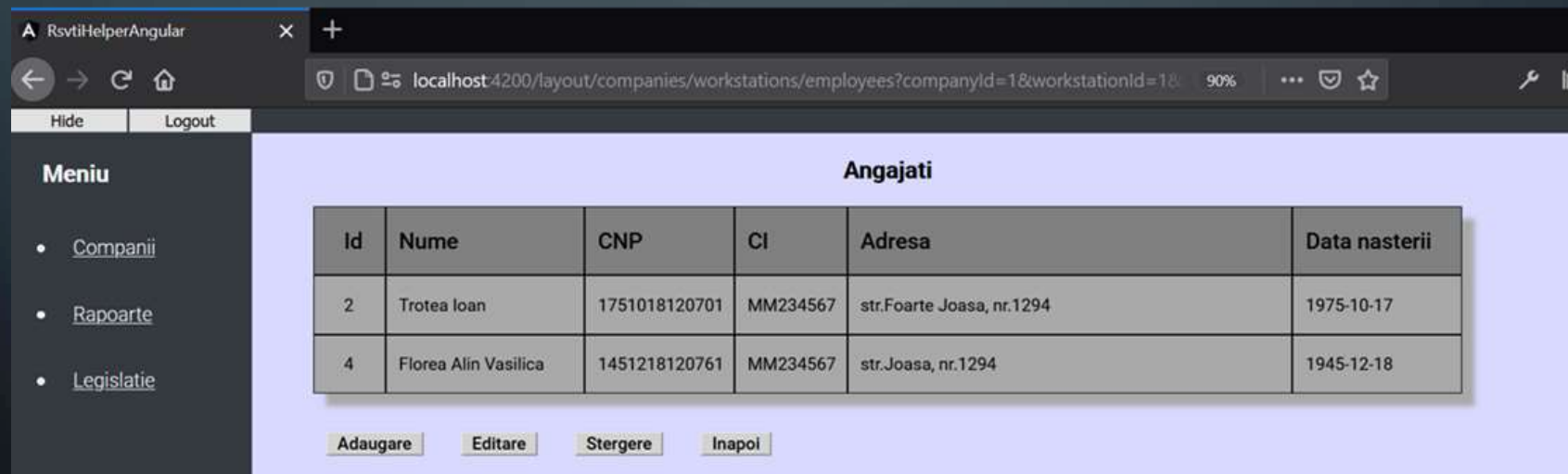
**Angajați** – afișează angajații corespunzători al unui punct de lucru selectat

**Instalații** – afișează instalațiile corespunzătoare care există într-un punct de lucru selectat



# Angajați

Se selectează un punct de lucru și se apasă butonul **Angajați** pentru afișarea angajaților punctului de lucru selectat.



# Afișarea angajaților

HideLogout

Meniu

- Companii
- Rapoarte
- Legislatie

Angajati

Id	Nume	CNP	CI	Adresa	Data nasterii
2	Trotea Ioan	1751018120001	MM234567	str.Foarte Joasa, nr.1294	1975-10-17
4	Florea Alin VASILICA	1451218120761	MM234567	str.Joasa, nr.1294	1945-12-18

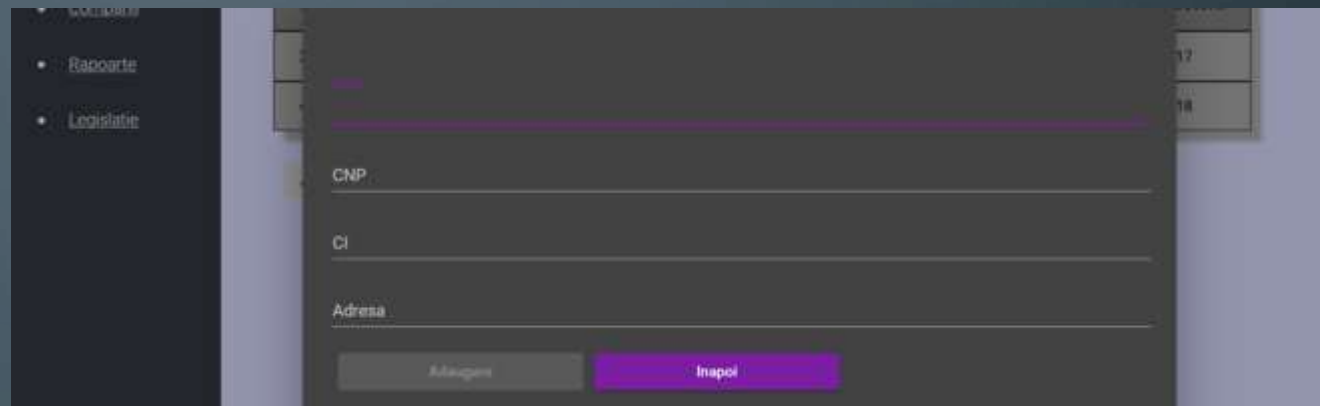
AdaugareEditareStergereAutorizareInapoi

Pagina **Angajați** conține:

- Tabelul cu angajații punctului de lucru selectat
- Butoane
  - **Adăugare** – adaugă un nou angajat în baza de date
  - **Editare** – editează datele unui angajat existent
  - **Ștergere** – șterge un angajat din baza de date
  - **Autorizare** – autorizează anumiți angajați sau introduce autorizații, dacă există
  - **Înapoi** – afișează din nou pagina **Puncte de lucru**

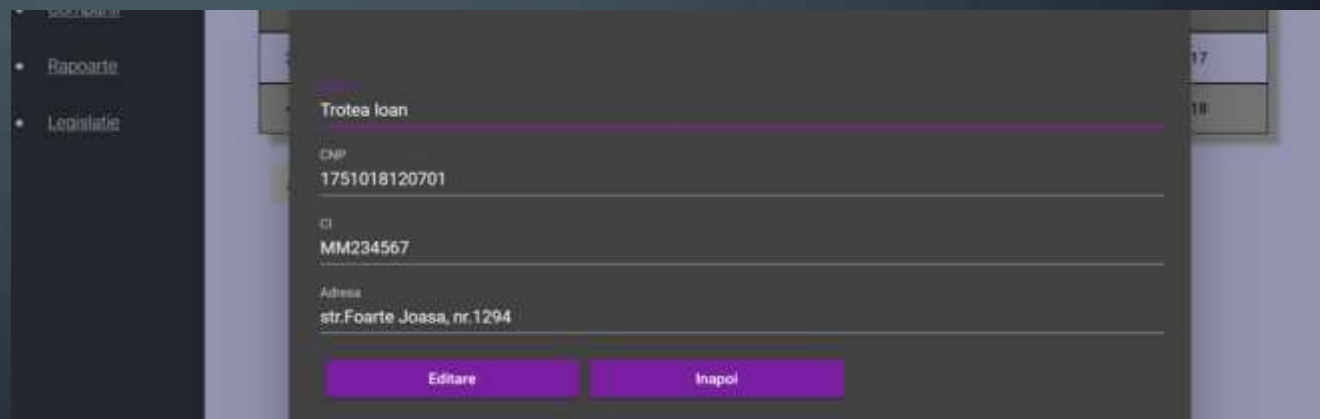
## Ferestrele de adăugare și editare a unui angajat

Butonul **Adăugare** deschide o fereastră tip dialog pentru adăugarea unui angajat nou în baza de date.



This screenshot shows a dark-themed dialog window for adding a new employee. On the left, a sidebar contains a menu with 'Raportare' and 'Legislație'. The main area of the dialog has a title bar with a red close button. Below the title bar, there are four input fields labeled 'Nume', 'CNP', 'CI', and 'Adresa'. At the bottom of the dialog, there are two buttons: 'Adăugare' (Add) in a light blue color and 'Înapoi' (Back) in a red color.

Butonul **Editare** deschide o fereastră de tip dialog care conține datele angajatului selectat pentru editare.



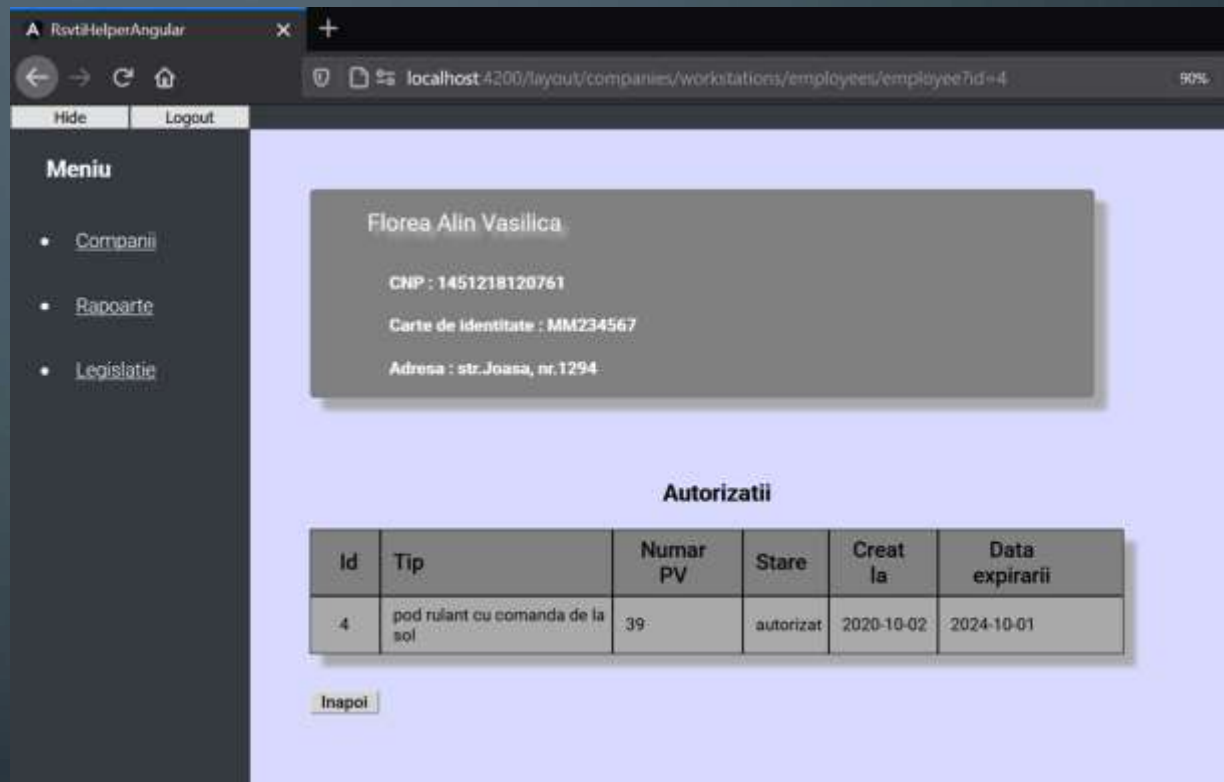
This screenshot shows a dark-themed dialog window for editing an existing employee. The layout is similar to the 'Adăugare' window, with a sidebar menu on the left. The main area has a title bar with a red close button. Below the title bar, the fields are pre-filled with data for an employee named 'Trotea Ioan': 'CNP' is '1751018120701', 'CI' is 'MM234567', and 'Adresa' is 'str.Foarte Joasa, nr.1294'. At the bottom, there are two buttons: 'Editare' (Edit) in a light blue color and 'Înapoi' (Back) in a red color.

# Afișare date angajat și autorizații deținute de acesta

Se deschide cu dublu click pe un angajat din tabelul cu **Angajați** și afișează:

- Datele personale ale angajatului

- Un tabel cu toate autorizațiile pe care le are angajatul



The screenshot shows a web browser window with the URL `localhost:4200/layout/companies/workstations/employees/employee?id=4`. The application has a dark sidebar menu with options: **Meniu**, [Companii](#), [Rapoarte](#), and [Legislație](#). The main content area displays the details for employee Florea Alin Vasilica, including their CNP (1451218120761), ID card number (MM234567), and address (str. Joasa, nr. 1294). Below this, a section titled **Autorizatii** contains a table with one authorization record.

Id	Tip	Numar PV	Stare	Creat la	Data expirării
4	pod rulant cu comanda de la sol	39	autorizat	2020-10-02	2024-10-01

An [Inapoi](#) button is located at the bottom left of the main content area.



# Afișarea instalațiilor

Se afișează la selectarea unui punct de lucru și apăsarea butonului **Instalații** din pagina **Puncte de lucru**

## Conține:

- Un tabel cu Instalațiile de ridicat din acel punct de lucru
- Un tabel cu Instalațiile sub presiune din acel punct de lucru.
- Buton de **Adăugare** instalație

*Fiecare instalație are buton de editare și ștergere*

HideLogout

Meniu

- Companii
- Rapoarte
- Legislatie

Puncte de lucru

Id	Adresa punctului de lucru
1	str.Parcului nr.43, loc Prundis, jud. Maramures
7	str. Stefan cel Mare, nr.1493,loc Suceava, jud. Suceava

AdaugareEditareStergereInapoi

AngajatiInstalatii

HideLogout

Meniu

- Companii
- Rapoarte
- Legislatie

Instalatii de ridicat

Id	Tip	Producator	Model	Serie	Anul fabricatiei	Numar ISCIR	Data expirarii	Editare	Stergere
1	Macara-automacara	Demag	C-100	5790SDEM	2014	SBMCR123875	2024-01-01	Editaza	Sterge
17	Macara-automacara	Demag	C-12	57999DEM	2018	SBMCR14325	2024-10-01	Editaza	Sterge

Instalatii sub presiune

Id	Tip	Producator	Model	Serie	Anul fabricatiei	Numar ISCIR	Data expirarii	Editare	Stergere
1	Recipient-pentru lichide din grupa 2 (lichide nepericuloase)	Cemb	CM-152634	9684976	2012	SBPS-2863	2022-10-10	Editaza	Sterge
6	Recipient-pentru lichide din grupa 2 (lichide nepericuloase)	Cemb	CM-152634	85476876	2012	SBPS-2023	2028-10-10	Editaza	Sterge

AdaugaInapoi

# Editarea datelor unei instalații

La selectarea celulei **Editează** se afișează o fereastră de tip dialog în care :

- în partea stânga sunt afișate valorile editabile comune tuturor instalațiilor
- în partea dreaptă sunt afișate valorile editabile particulare fiecărei instalații după categorie și subcategorie
- butonul **Editare** – salvează datele noi în baza de date al aceluși ID
- butonul **Înapoi** – închide fereastra de editare

The screenshot shows a software interface for editing crane data. The dialog box is titled "Macara-automacara" and contains two columns of input fields. The left column lists common attributes for all cranes, and the right column lists specific attributes for the selected crane. At the bottom right, there are two buttons: "Editare" (Edit) and "Inapoi" (Back).

Macara-automacara	
Sarcina nominala(kg)	120000
Inaltime de ridicare (m)	25
Raza (m)	25
Serie de fabricatie	5790SDEM
Numar ISCIR	SBMCR123875
Anul fabricatiei	2011
Producator	Demag
Model	C-100
Sarcina nominala pe carligul auxiliar (kg)	5000
Companie intretinatoare	DEME
	manual
<b>Editare</b> <b>Inapoi</b>	

# Afișarea detaliilor instalației și autorizările acesteia

Se afișează după un dublu click pe instalația dorită din tabelul **Instalațiilor**

Afișează:

- Toate datele particulare ale instalației selectate
- Un tabel cu toate autorizațiile pe care le are instalația selectată

The screenshot shows a web browser window with the address bar displaying 'localhost:4200/layout/companies/w'. The application has a dark sidebar with a 'Meniu' (Menu) section containing links for 'Companii', 'Raporte', and 'Legislatie'. The main content area displays details for an 'Instalatie sub presiune' (Pressurized installation). The details include: Denumire: Recipient- pentru lichide din grupa 2 (lichide nepericuloase), Tip: CM-152634, Numar ISCIR: SBPS-28639898, Serie: 9684976, Producator: Comb, Temperatura maxima de functionare: 120 grC, Temperatura minima de functionare: -10 grC, Presiune nominala: 12 bari, Volum: 1500 L, Tip fluid: aer, Anul fabricatiei: 2012, Loc de functionare: Suprateran, Data expirarii: 2022-10-10, and Compania de intretinere: Da. Below the details is a table titled 'Autorizatii' (Authorizations) with columns: Id, Tip, Numar PV, Stare, Creat la, and Data expirarii. The table contains one row with the following data: Id: 6, Tip: (empty), Numar PV: 2, Stare: autorizat, Creat la: 2020-10-04, and Data expirarii: 2021-10-04.

Instalatie sub presiune

Denumire: Recipient- pentru lichide din grupa 2 (lichide nepericuloase)

Tip : CM-152634

Numar ISCIR : SBPS-28639898

Serie : 9684976

Producator : Comb

Temperatura maxima de functionare : 120 grC

Temperatura minima de functionare : -10 grC

Presiune nominala : 12 bari

Volum : 1500 L

Tip fluid : aer

Anul fabricatiei : 2012

Loc de functionare : Suprateran

Data expirarii : 2022-10-10

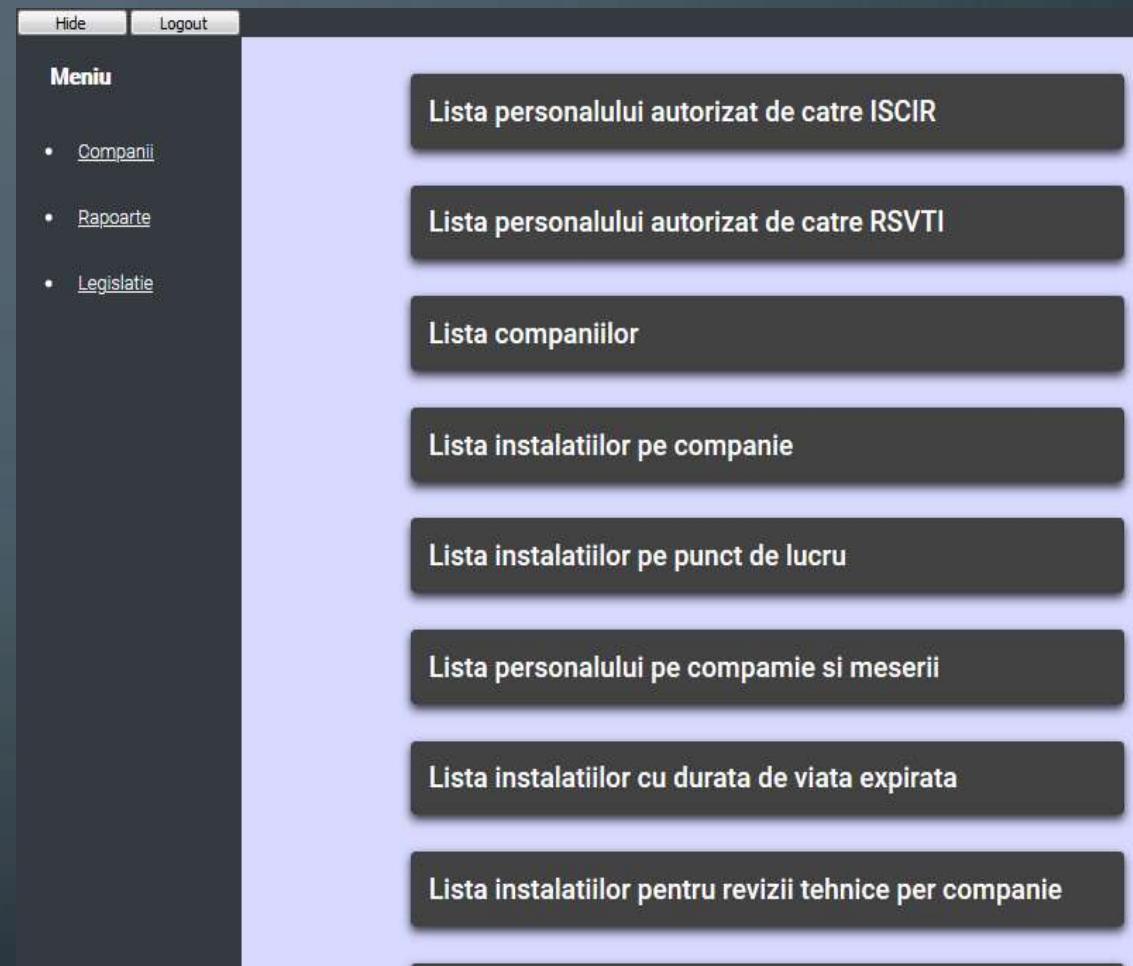
Compania de intretinere : Da

**Autorizatii**

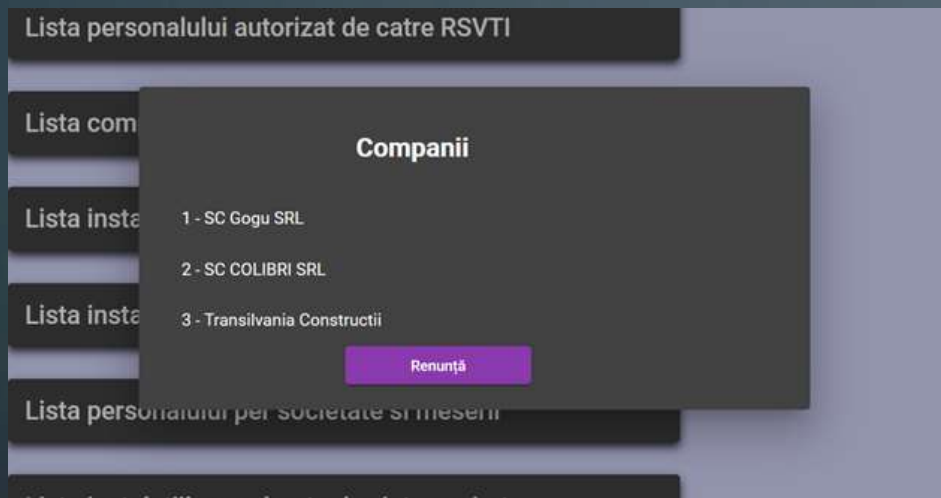
Id	Tip	Numar PV	Stare	Creat la	Data expirarii
6		2	autorizat	2020-10-04	2021-10-04

# Rapoarte

La selectarea din **Meniu** a rapoartelor, se afișează o listă cu rapoartele utile pe care aplicația le poate genera, cu valorile din baza de date.

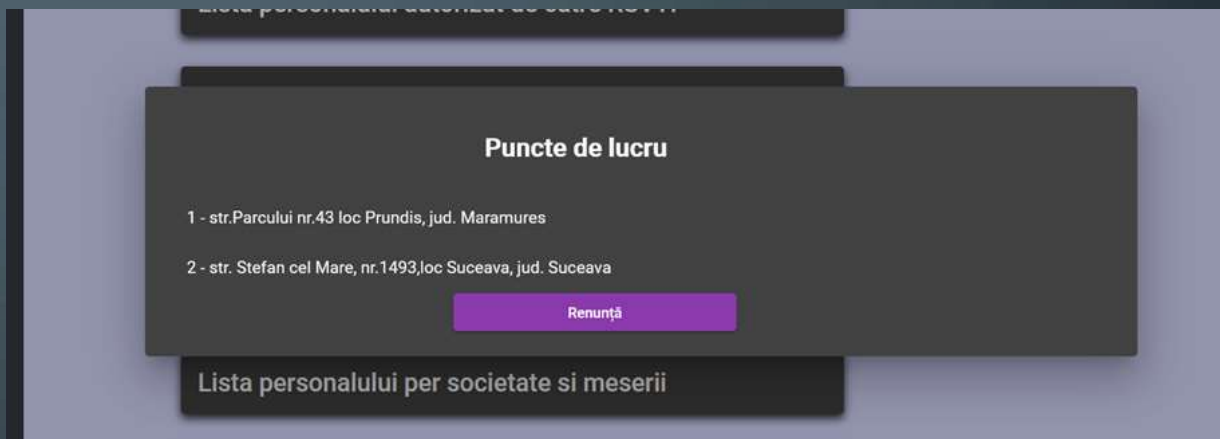


# Rapoarte



Apoi apare și o listă cu punctele de lucru ale companiei selectate, în prima fereastră, din care se alege punctul de lucru pentru care se dorește raportul.

După selectarea tipului de raport dorit, după caz, apare o fereastră de tip dialog cu lista companiilor din care să se aleagă compania pentru care se dorește raportul.



Butonul **Renunță** din fiecare fereastră dă posibilitatea renunțării înainte de generarea raportului.





# Rapoarte – metoda în back-end pentru generarea documentului .docx

```
41
42 public String getAllEmployeesISCIRAuthorization(Long id1, Long id2, String type) throws IOException {
43     String authorizationType;
44     if (type.equals("ISCIR")) {...} else {...}
45     LocalDateTime dateTime = LocalDateTime.now();
46     DateTimeFormatter dateTimeFormatter = DateTimeFormatter.ofPattern("dd-MM-yyyy-HH-mm-ss");
47     String date = dateTime.format(dateTimeFormatter);
48     Company company = companyService.getCompanyById(id1);
49     Workstation workstation = workstationService.getWorkstationById(id2);
50
51     XMPFDocument document = new XMPFDocument();
52     String documentPath = "/Lista personalului autorizat de catre " + type + " "
53         + date + ".docx";
54     String directory = "D:/Documente salvate";
55
56     File directoryFile = new File(directory);
57     if (!directoryFile.exists()) {
58         directoryFile.mkdir();
59     }
60
61     FileOutputStream out = new FileOutputStream(
62         new File(pathname) directoryFile + documentPath));
63     nameAndAddressCompanyLandscape(company, company.getCompanyName(), workstation.getAddress(), document);
64
65     titleTable(document, "Lista personalului autorizat de catre " + type);
66
67     XMPFTable table = document.createTable();
68
69     XMPFTableRow headerTableRow = table.getRow(pos: 0);
70     headerTableRow.getCell(pos: 0).setText("Id");
```

```
70
71 XMPFTable table = document.createTable();
72
73 XMPFTableRow headerTableRow = table.getRow(pos: 0);
74 headerTableRow.getCell(pos: 0).setText("Id");
75 headerTableRow.addNewTableCell().setText("Nume");
76 headerTableRow.addNewTableCell().setText("CNP");
77 headerTableRow.addNewTableCell().setText("Meseria ");
78 headerTableRow.addNewTableCell().setText("Instalatie deservita ");
79 headerTableRow.addNewTableCell().setText("Serie Autorizatie ");
80 headerTableRow.addNewTableCell().setText("Data expirarii talonului ");
81 headerTableRow.addNewTableCell().setText("Data expirarii vizei ");
82 headerTableRow.addNewTableCell().setText("Observatii ");
83
84 List<Employee> employeeList = employeeService.getAllByWorkstation(workstation.getWorkstationId());
85
86 for (Employee employee : employeeList) {
87     List<Authorization> authorizationList = authorizationService
88         .getAllByEmployeeIdAndType(employee.getEmployeeId(), authorizationType);
89     allEmployees(table, employee, authorizationList);
90 }
91
92 signatureParagraph1(document, company);
93 document.write(out);
94 out.close();
95 return documentPath;
96 }
```



**Raportul** se salvează local și este un fișier **.docx** care poate fi deschis și editat mai departe, dacă se dorește.

Name	Date modified	Type	Size
 Lista personalului autorizat de catre ISCIR 19_10_2020_19_37_11.docx	19.10.2020 19:37	Microsoft Word D...	3 KB
 Lista personalului autorizat de catre RSVTI 11_10_2020_11_11_45.docx	11.10.2020 11:11	Microsoft Word D...	3 KB
 Proces verbal persoane_02_10_2020_19_12_02.docx	02.10.2020 19:12	Microsoft Word D...	4 KB
 Proces verbal verificare tehnica_04_10_2020_13_35_20.docx	04.10.2020 13:35	Microsoft Word D...	4 KB

SC Gogu SRL  
str.Mare, nr.543, loc. Cluj-Napoca, jud. Cluj  
Punct de lucru:  
str.Parcului nr.43 loc Prundis, jud. Maramures.

**Lista personalului autorizat de catre ISCIR**

Id	Nume	CNP	Meseria	Instalatie deservita	Serie Autorizatie	Data expirarii talonului	Data expirarii vizei	Observatii
2	Trotea Ioan	17510181207 01	Stivulatorist	stivuator autopropulsat	SCte5463	2020-10-02	2024-10-01	
4	Florea Alin Vasilica	14512181207 61	Macaragiu categoria E	pod rulant cu comanda de la sol	FUIG87687632	2020-10-02	2024-10-01	

**DIRECTOR**  
Vasile De Peste Deal

**RSVTI**  
Baciu Dan Raul

Page 1 of 1 | 76 words, 540 characters | Default Style | English (UK)

The image features a dark blue gradient background. In the corners, there are decorative white line art elements resembling circuit boards or neural network connections, with small circles at the end of the lines.

Vă mulțumesc!