שלי API אני רוצה שתעזור לי להכין את ה

:מה אני צריכה להכין בשבילה בדוטנט SERVICEזה האנגולר תגיד לי ממש כל פונקציה ב

```
import { Component, signal, inject, OnInit, OnDestroy } from '@angular/core';

import { CommonModule } from '@angular/common';

import { FormsModule } from '@angular/forms';

import { LucideAngularModule } from 'lucide-angular';

import { Subject, takeUntil } from 'rxjs';

import { ApiService } from '../../services/api.service.service';

import { Category } from '../../models/api.models';

@Component({

selector: 'app-categories',

standalone: true,

imports: [CommonModule, FormsModule, LucideAngularModule],

templateUrl: './categories.component.html',

styleUrls: ['./categories.component.css']

})

export class CategoriesComponent implements OnInit, OnDestroy {

apiService = inject(ApiService);

private destroy$ = new Subject<void>();

categories = signal<Category[]>([]);

showCategoryModal = signal(false);

editingCategory = signal<Category | null>(null);

categoryFormData: Partial<Category> = {

name: '',

description: '',
isActive: true
};
ngOnInit() {
this.loadCategories();
}
ngOnDestroy() {
this.destroy$.next();
this.destroy$.complete();
}
loadCategories() {
this.apiService.getCategories()
.pipe(takeUntil(this.destroy$))
.subscribe({
next: (categories) => {
this.categories.set(categories);
},
error: (error) => {
console.error('Error loading categories:', error);
}
});
}
openCreateCategoryModal() {
this.editingCategory.set(null);
this.categoryFormData = {
name: '',
description: '',
isActive: true
};
this.showCategoryModal.set(true);
}
editCategory(category: Category) {
this.editingCategory.set(category);
this.categoryFormData = {
name: category.name,
description: category.description,
isActive: category.isActive
};
this.showCategoryModal.set(true);
}
closeCategoryModal() {
this.showCategoryModal.set(false);
this.editingCategory.set(null);
}
saveCategory() {
const editingCategory = this.editingCategory();
if (editingCategory) {
this.apiService.updateCategory(editingCategory.id, this.categoryFormData)
.pipe(takeUntil(this.destroy$))
.subscribe({
next: () => {
```

```
this.closeCategoryModal();
this.loadCategories();
},
error: (error) => {
console.error('Error updating category:', error);
}
});
} else {
this.apiService.createCategory(this.categoryFormData)
.pipe(takeUntil(this.destroy$))
.subscribe({
next: () => {
this.closeCategoryModal();
this.loadCategories();
},
error: (error) => {
console.error('Error creating category:', error);
}
});
}
}
deleteCategory(categoryId: number) {
const category = this.categories().find(c => c.id === categoryId);
if (category && category.fileCount > 0) {
alert(`לא ניתן למחוק קטגוריה שמכילה ${category.fileCount} קבצים. העבר תחילה את הקבצים
לקטגוריה אחרת.`);
```

```
return;
}
if (!confirm('האם אתה בטוח שברצונך למחוק את הקטגוריה? פעולה זו לא ניתנת לביטול.')) {
return;
}
this.apiService.deleteCategory(categoryId)
.pipe(takeUntil(this.destroy$))
.subscribe({
next: () => {
this.loadCategories();
},
error: (error) => {
console.error('Error deleting category:', error);
}
});
}
trackByCategoryId(index: number, category: Category): number {
return category.id;
}
formatDate(dateString: string): string {
const date = new Date(dateString);
return date.toLocaleDateString('he-IL');
}
}
<div class="space-y-6">
```

```
<!-- Header -->
<div class="flex justify-between items-center">
<h2 class="text-xl font-bold">ניהול קטגוריות</h2>
<button
(click)="openCreateCategoryModal()"
class="px-4 py-2 bg-gradient-to-r from-orange-500 to-yellow-500 hover:from-orange-600 hover:to-yellow-600 text-white rounded-lg flex items-center"
>
<i-lucide name="plus" class="w-4 h-4 ml-2"></i-lucide>
 הוסף קטגוריה
</button>
</div>
<!-- Categories Grid -->
<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
@for (category of categories(); track trackByCategoryId($index, category)) {
<div class="border-0 shadow-lg bg-white/80 backdrop-blur-sm rounded-lg p-6 hover:shadow-xl transition-all duration-200">
<div class="flex items-start justify-between mb-4">
<div class="flex items-center">
<div class="p-3 rounded-lg bg-gradient-to-br from-orange-100 to-yellow-100 ml-3">
<i-lucide name="folder-open" class="h-6 w-6 text-orange-600"></i-lucide>
</div>
<div>
<h3 class="text-lg font-semibold text-gray-900">{{ category.name }}</h3>
<p class="text-sm text-gray-500">{{ category.fileCount }} קבצים</p>
</div>
```

```
</div>
<div class="flex items-center space-x-1 space-x-reverse">
<button
(click)="editCategory(category)"
class="p-2 text-gray-400 hover:text-blue-600 hover:bg-blue-50 rounded-lg transition-all duration-200"
title="ערוך קטגוריה"
>
<i-lucide name="edit" class="h-4 w-4"></i-lucide>
</button>
<button
(click)="deleteCategory(category.id)"
class="p-2 text-gray-400 hover:text-red-600 hover:bg-red-50 rounded-lg transition-all duration-200"
title="מחק קטגוריה"
>
<i-lucide name="trash-2" class="h-4 w-4"></i-lucide>
</button>
</div>
</div>
@if (category.description) {
<div class="mb-4">
<p class="text-sm text-gray-600">{{ category.description }}</p>
</div>
}
<div class="flex items-center justify-between text-xs text-gray-500">
```

```
<span>נוצר: {{ formatDate(category.createdAt) }}</span>
<span
class="px-2 py-1 rounded-full"
[ngClass]="{
'bg-green-100 text-green-800': category.isActive,
'bg-red-100 text-red-800': !category.isActive
}"
>
{{ category.isActive ? 'פעיל' : 'לא פעיל' }}
</span>
</div>
</div>
} @empty {
<div class="col-span-full">
<div class="text-center py-12">
<i-lucide name="folder-open" class="mx-auto h-16 w-16 mb-4 text-gray-300"></i-lucide>
<h3 class="text-lg font-medium text-gray-900 mb-2">אין קטגוריות</h3>
<p class="text-gray-500 mb-4">התחל על ידי יצירת קטגוריה ראשונה</p>
<button
(click)="openCreateCategoryModal()"
class="px-4 py-2 bg-gradient-to-r from-orange-500 to-yellow-500 hover:from-orange-600 hover:to-yellow-600 text-white rounded-lg"
>
 צור קטגוריה
</button>
```

```
</div>

</div>

}

</div>

<!-- Create/Edit Category Modal -->

@if (showCategoryModal()) {

<div class="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50">

<div class="bg-white rounded-lg p-6 w-full max-w-md mx-4">

<div class="flex justify-between items-center mb-4">

<h3 class="text-lg font-semibold">

{{ editingCategory() ? 'ערוך קטגוריה' : 'הוסף קטגוריה חדשה' }}

</h3>

<button (click)="closeCategoryModal()" class="text-gray-400 hover:text-gray-600">

<i-lucide name="x" class="h-5 w-5"></i-lucide>

</button>

</div>

<form (ngSubmit)="saveCategory()" #categoryForm="ngForm">

<div class="space-y-4">

<div>

<label class="block text-sm font-medium text-gray-700 mb-1">שם הקטגוריה</label>

<input

type="text"

[(ngModel)]="categoryFormData.name"

name="name"

required
```

```
class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-orange-500 focus:ring-1 focus:ring-orange-500"

placeholder="הכנס שם קטגוריה"

/>

</div>

<div>

<label class="block text-sm font-medium text-gray-700 mb-1"> תיאור (אופציונלי)</label>

<textarea

[(ngModel)]="categoryFormData.description"

name="description"

rows="3"

class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-orange-500 focus:ring-1 focus:ring-orange-500"

placeholder="הכנס תיאור לקטגוריה"

></textarea>

</div>

<div class="flex items-center">

<input

type="checkbox"

[(ngModel)]="categoryFormData.isActive"

name="isActive"

id="categoryIsActive"

class="ml-2"

/>

<label for="categoryIsActive" class="text-sm text-gray-700">קטגוריה פעילה</label>

</div>
```

```
</div>
<div class="flex justify-end space-x-2 space-x-reverse mt-6">
<button
type="button"
(click)="closeCategoryModal()"
class="px-4 py-2 border border-gray-300 rounded-lg hover:bg-gray-50"
>
ביטול
</button>
<button
type="submit"
[disabled]="!categoryForm.valid || apiService.loading()"
class="px-4 py-2 bg-gradient-to-r from-orange-500 to-yellow-500 hover:from-orange-600 hover:to-yellow-600 text-white rounded-lg disabled:opacity-50"
>
{{ editingCategory() ? 'הוסף' : 'עדכן' }}
</button>
</div>
</form>
</div>
</div>
}
<!-- Loading State -->
@if (apiService.loading()) {
<div class="text-center py-8">
```

```
<div class="inline-flex items-center px-4 py-2 font-semibold leading-6 text-sm shadow rounded-md text-white bg-orange-500">

<svg class="animate-spin -ml-1 mr-3 h-5 w-5 text-white" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24">

<circle class="opacity-25" cx="12" cy="12" r="10" stroke="currentColor" stroke-width="4"></circle>

<path class="opacity-75" fill="currentColor" d="M4 12a8 8 0 018-8V0C5.373 0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0 014 12H0c0 3.042 1.135 5.824 3 7.938l3-2.647z"></path>

</svg>

טוען קטגוריות...

</div>

</div>

}

<!-- Error State -->

@if (apiService.error()) {

<div class="bg-red-50 border border-red-200 rounded-lg p-4">

<div class="flex items-center">

<i-lucide name="alert-circle" class="h-5 w-5 text-red-500 ml-2"></i-lucide>

<span class="text-red-700">{{ apiService.error() }}</span>

</div>

</div>

}

</div>

import { Component, signal, inject, OnInit, OnDestroy } from '@angular/core';

import { CommonModule } from '@angular/common';

import { LucideAngularModule } from 'lucide-angular';

import { Subject, takeUntil } from 'rxjs';
import { ApiService } from '../../services/api.service.service';

import { Stats } from '../../models/api.models';

@Component({

selector: 'app-dashboard',

standalone: true,

imports: [CommonModule, LucideAngularModule],

templateUrl: './dashboard.component.html',

styleUrls: ['./dashboard.component.css']

})

export class DashboardComponent implements OnInit, OnDestroy {

apiService = inject(ApiService);

private destroy$ = new Subject<void>();

stats = signal<Stats | null>(null);

ngOnInit() {

this.loadStats();

}

ngOnDestroy() {

this.destroy$.next();

this.destroy$.complete();

}

loadStats() {

this.apiService.getStats()

.pipe(takeUntil(this.destroy$))

.subscribe({

next: (data) => {
```

```
this.stats.set(data);

},

error: (error) => {

console.error('Error loading stats:', error);

}

});

}

formatBytes(bytes: number): string {

if (bytes === 0) return '0 Bytes';

const k = 1024;

const sizes = ['Bytes', 'KB', 'MB', 'GB', 'TB'];

const i = Math.floor(Math.log(bytes) / Math.log(k));

return parseFloat((bytes / Math.pow(k, i)).toFixed(2)) + ' ' + sizes[i];

}

getStoragePercentage(): number {

const stats = this.stats();

if (!stats || !stats.storageLimit) return 0;

return Math.round((stats.storageUsed / stats.storageLimit) * 100);

}

getLastBackupTime(): string {

return 'לפני 2 שעות';

}

}

<main class="p-6">

<!-- Stats Cards -->
```

```
<div class="grid grid-cols-1 md:grid-cols-4 gap-6 mb-8 fade-in-up">

<!-- Total Users Card -->

<div class="border-0 shadow-lg bg-gradient-to-br from-purple-500 to-purple-600 text-white overflow-hidden relative rounded-lg p-6">

<div class="absolute top-0 right-0 w-20 h-20 bg-white/10 rounded-full -mr-10 -mt-10"></div>

<div class="flex items-center justify-between mb-2 relative z-10">

<h3 class="text-sm font-medium text-purple-100">סך הכל משתמשים</h3>

<i-lucide name="users" class="h-6 w-6 text-white"></i-lucide>

</div>

<div class="relative z-10">

<div class="text-3xl font-bold">

{{ stats()?.totalUsers || 'טוען...' }}

</div>

<p class="text-xs text-purple-200">

פעילים: {{ stats()?.activeUsers || '...' }}

</p>

</div>

</div>

<!-- Total Files Card -->

<div class="border-0 shadow-lg bg-gradient-to-br from-green-500 to-emerald-600 text-white overflow-hidden relative rounded-lg p-6">

<div class="absolute top-0 right-0 w-20 h-20 bg-white/10 rounded-full -mr-10 -mt-10"></div>

<div class="flex items-center justify-between mb-2 relative z-10">

<h3 class="text-sm font-medium text-green-100">סך הכל קבצים</h3>

<i-lucide name="file-text" class="h-6 w-6 text-white"></i-lucide>
```

```
</div>
<div class="relative z-10">
<div class="text-3xl font-bold">
{{ stats()?.totalFiles || 'טוען...' }}
</div>
<p class="text-xs text-green-200">
מאושרים: {{ stats()?.approvedFiles || '...' }}
</p>
</div>
</div>
<!-- Downloads Card -->
<div class="border-0 shadow-lg bg-gradient-to-br from-orange-500 to-yellow-500 text-white overflow-hidden relative rounded-lg p-6">
<div class="absolute top-0 right-0 w-20 h-20 bg-white/10 rounded-full -mr-10 -mt-10"></div>
<div class="flex items-center justify-between mb-2 relative z-10">
<h3 class="text-sm font-medium text-orange-100">הורדות</h3>
<i-lucide name="download" class="h-6 w-6 text-white"></i-lucide>
</div>
<div class="relative z-10">
<div class="text-3xl font-bold">
{{ stats()?.totalDownloads || 'טוען...' }}
</div>
<p class="text-xs text-orange-200">
היום: {{ stats()?.todayDownloads || '...' }}
</p>
```

```
</div>
</div>
<!-- Pending Files Card -->
<div class="border-0 shadow-lg bg-gradient-to-br from-pink-500 to-rose-600 text-white overflow-hidden relative rounded-lg p-6">
<div class="absolute top-0 right-0 w-20 h-20 bg-white/10 rounded-full -mr-10 -mt-10"></div>
<div class="flex items-center justify-between mb-2 relative z-10">
<h3 class="text-sm font-medium text-pink-100">ממתינים לאישור</h3>
<i-lucide name="user-check" class="h-6 w-6 text-white"></i-lucide>
</div>
<div class="relative z-10">
<div class="text-3xl font-bold">
{{ stats()?.pendingFiles || 'טוען...' }}
</div>
<p class="text-xs text-pink-200">דורש טיפול</p>
</div>
</div>
</div>
<!-- Storage Usage -->
<div class="grid grid-cols-1 lg:grid-cols-2 gap-6 mb-8">
<div class="border-0 shadow-lg bg-white/80 backdrop-blur-sm rounded-lg p-6">
<h3 class="text-lg font-semibold mb-4 flex items-center">
<i-lucide name="hard-drive" class="ml-2 h-5 w-5 text-blue-600"></i-lucide>
 שימוש באחסון
</h3>
```

```
<div class="space-y-2">
<div class="flex justify-between text-sm">
<span>בשימוש</span>
<span>{{ formatBytes(stats()?.storageUsed || 0) }}</span>
</div>
<div class="w-full bg-gray-200 rounded-full h-2">
<div
class="bg-gradient-to-r from-blue-500 to-purple-500 h-2 rounded-full transition-all duration-300"
[style.width.%]="getStoragePercentage()"
></div>
</div>
<div class="flex justify-between text-xs text-gray-500">
<span>{{ getStoragePercentage() }}% בשימוש</span>
<span>מתוך {{ formatBytes(stats()?.storageLimit || 0) }}</span>
</div>
</div>
</div>
<div class="border-0 shadow-lg bg-white/80 backdrop-blur-sm rounded-lg p-6">
<h3 class="text-lg font-semibold mb-4 flex items-center">
<i-lucide name="activity" class="ml-2 h-5 w-5 text-green-600"></i-lucide>
 סטטוס מערכת
</h3>
<div class="space-y-3">
<div class="flex items-center justify-between">
<span class="text-sm">חיבור לשרת</span>
```

```
<span class="flex items-center text-green-600">
<div class="w-2 h-2 bg-green-500 rounded-full ml-2 animate-pulse"></div>
 מחובר
</span>
</div>
<div class="flex items-center justify-between">
<span class="text-sm">מסד נתונים</span>
<span class="flex items-center text-green-600">
<div class="w-2 h-2 bg-green-500 rounded-full ml-2"></div>
 פעיל
</span>
</div>
<div class="flex items-center justify-between">
<span class="text-sm">גיבוי אחרון</span>
<span class="text-sm text-gray-500">{{ getLastBackupTime() }}</span>
</div>
</div>
</div>
</div>
<!-- Loading State -->
@if (apiService.loading()) {
<div class="text-center py-8">
<div class="inline-flex items-center px-4 py-2 font-semibold leading-6 text-sm shadow rounded-md text-white bg-purple-500">
<svg class="animate-spin -ml-1 mr-3 h-5 w-5 text-white" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24">
<circle class="opacity-25" cx="12" cy="12" r="10" stroke="currentColor" stroke-width="4"></circle>

<path class="opacity-75" fill="currentColor" d="M4 12a8 8 0 018-8V0C5.373 0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0 014 12H0c0 3.042 1.135 5.824 3 7.938l3-2.647z"></path>

</svg>

טוען נתונים...

</div>

</div>

}

<!-- Error State -->

@if (apiService.error()) {

<div class="bg-red-50 border border-red-200 rounded-lg p-4 mb-6">

<div class="flex items-center">

<i-lucide name="alert-circle" class="h-5 w-5 text-red-500 ml-2"></i-lucide>

<span class="text-red-700">{{ apiService.error() }}</span>

</div>

</div>

}

</main>

import { Component, signal, inject, OnInit, OnDestroy } from '@angular/core';

import { CommonModule } from '@angular/common';

import { FormsModule } from '@angular/forms';

import { LucideAngularModule } from 'lucide-angular';

import { Subject, takeUntil } from 'rxjs';

import { ApiService } from '../../services/api.service.service';

import { FileItem, Category, ApiResponse } from '../../models/api.models';
```

```
@Component({
selector: 'app-files',
standalone: true,
imports: [CommonModule, FormsModule, LucideAngularModule],
templateUrl: './files.component.html',
styleUrls: ['./files.component.css']
})
export class FilesComponent implements OnInit, OnDestroy {
apiService = inject(ApiService);
private destroy$ = new Subject<void>();
files = signal<FileItem[]>([]);
categories = signal<Category[]>([]);
pagination = signal<any>(null);
uploadProgress = signal(0);
selectedStatus = '';
selectedCategory = '';
currentPage = 1;
pageSize = 10;
ngOnInit() {
this.loadFiles();
this.loadCategories();
}
ngOnDestroy() {
this.destroy$.next();
this.destroy$.complete();
```

```
}
loadFiles() {
const categoryId = this.selectedCategory ? parseInt(this.selectedCategory) : undefined;
this.apiService.getFiles(this.currentPage, this.pageSize, this.selectedStatus, categoryId)
.pipe(takeUntil(this.destroy$))
.subscribe({
next: (response) => {
this.files.set(response.data);
this.pagination.set(response.pagination);
},
error: (error) => {
console.error('Error loading files:', error);
}
});
}
loadCategories() {
this.apiService.getCategories()
.pipe(takeUntil(this.destroy$))
.subscribe({
next: (categories) => {
this.categories.set(categories);
},
error: (error) => {
console.error('Error loading categories:', error);
}
});
}
onFilterChange() {
this.currentPage = 1;
this.loadFiles();
}
onFileSelect(event: any) {
const file = event.target.files[0];
if (!file) return;
if (!this.selectedCategory) {
alert('אנא בחר קטגוריה לפני העלאת הקובץ');
return;
}
this.uploadProgress.set(0);
this.apiService.uploadFile(file, parseInt(this.selectedCategory))
.pipe(takeUntil(this.destroy$))
.subscribe({
next: (uploadedFile) => {
this.uploadProgress.set(100);
setTimeout(() => {
this.uploadProgress.set(0);
this.loadFiles();
}, 1000);
},
error: (error) => {
console.error('Error uploading file:', error);
this.uploadProgress.set(0);
}
});
}
approveFile(fileId: number) {
this.apiService.approveFile(fileId)
.pipe(takeUntil(this.destroy$))
.subscribe({
next: () => {
this.loadFiles();
},
error: (error) => {
console.error('Error approving file:', error);
}
});
}
rejectFile(fileId: number) {
const reason = prompt('סיבת דחייה (אופציונלי):');
this.apiService.rejectFile(fileId, reason || undefined)
.pipe(takeUntil(this.destroy$))
.subscribe({
next: () => {
this.loadFiles();
},
error: (error) => {
console.error('Error rejecting file:', error);
}
});
}
deleteFile(fileId: number) {
if (!confirm('האם אתה בטוח שברצונך למחוק את הקובץ?')) {
return;
}
this.apiService.deleteFile(fileId)
.pipe(takeUntil(this.destroy$))
.subscribe({
next: () => {
this.loadFiles();
},
error: (error) => {
console.error('Error deleting file:', error);
}
});
}
downloadFile(fileId: number, fileName: string) {
this.apiService.downloadFile(fileId)
.pipe(takeUntil(this.destroy$))
.subscribe({
```

```
next: (blob) => {
const url = window.URL.createObjectURL(blob);
const link = document.createElement('a');
link.href = url;
link.download = fileName;
link.click();
window.URL.revokeObjectURL(url);
},
error: (error) => {
console.error('Error downloading file:', error);
}
});
}
previousPage() {
if (this.currentPage > 1) {
this.currentPage--;
this.loadFiles();
}
}
nextPage() {
const pagination = this.pagination();
if (pagination && this.currentPage < pagination.totalPages) {
this.currentPage++;
this.loadFiles();
}
```

```
}
trackByFileId(index: number, file: FileItem): number {
return file.id;
}
formatFileSize(bytes: number): string {
if (bytes === 0) return '0 Bytes';
const k = 1024;
const sizes = ['Bytes', 'KB', 'MB', 'GB'];
const i = Math.floor(Math.log(bytes) / Math.log(k));
return parseFloat((bytes / Math.pow(k, i)).toFixed(2)) + ' ' + sizes[i];
}
formatDate(dateString: string): string {
const date = new Date(dateString);
return date.toLocaleDateString('he-IL');
}
getStatusText(status: string): string {
switch (status) {
case 'pending': return 'ממתין לאישור';
case 'approved': return 'מאושר';
case 'rejected': return 'נדחה';
default: return status;
}
}
}
<div class="space-y-6">
```

```
<!-- Header -->

<div class="flex justify-between items-center">

<h2 class="text-xl font-bold">ניהול קבצים</h2>

<div class="flex space-x-2 space-x-reverse">

<select

[(ngModel)]="selectedStatus"

(change)="onFilterChange()"

class="px-3 py-2 border border-gray-300 rounded-lg focus:border-purple-500 focus:ring-1 focus:ring-purple-500"

>

<option value="">כל הסטטוסים</option>

<option value="pending">ממתין לאישור</option>

<option value="approved">מאושר</option>

<option value="rejected">נדחה</option>

</select>

<select

[(ngModel)]="selectedCategory"

(change)="onFilterChange()"

class="px-3 py-2 border border-gray-300 rounded-lg focus:border-purple-500 focus:ring-1 focus:ring-purple-500"

>

<option value="">כל הקטגוריות</option>

@for (category of categories(); track category.id) {

<option [value]="category.id">{{ category.name }}</option>

}

</select>
```

```
<button
(click)="loadFiles()"
class="px-4 py-2 border border-gray-300 rounded-lg hover:bg-gray-50 flex items-center"
>
<i-lucide name="refresh-cw" class="w-4 h-4 ml-2"></i-lucide>
 רענן
</button>
<label class="px-4 py-2 bg-gradient-to-r from-purple-500 to-blue-500 hover:from-purple-600 hover:to-blue-600 text-white rounded-lg cursor-pointer flex items-center">
<i-lucide name="upload" class="w-4 h-4 ml-2"></i-lucide>
 העלה קובץ
<input
type="file"
class="hidden"
(change)="onFileSelect($event)"
accept=".pdf,.doc,.docx,.ppt,.pptx,.xls,.xlsx,.txt,.jpg,.jpeg,.png,.gif"
/>
</label>
</div>
</div>
<!-- Upload Progress -->
@if (uploadProgress() > 0 && uploadProgress() < 100) {
<div class="bg-blue-50 border border-blue-200 rounded-lg p-4">
<div class="flex items-center justify-between mb-2">
<span class="text-sm font-medium text-blue-700">מעלה קובץ...</span>
<span class="text-sm text-blue-600">{{ uploadProgress() }}%</span>
</div>
<div class="w-full bg-blue-200 rounded-full h-2">
<div
class="bg-blue-500 h-2 rounded-full transition-all duration-300"
[style.width.%]="uploadProgress()"
></div>
</div>
</div>
}
<!-- Files Table -->
<div class="border-0 shadow-lg bg-white/80 backdrop-blur-sm rounded-lg overflow-hidden">
<div class="overflow-x-auto">
<table class="w-full">
<thead class="bg-gradient-to-r from-purple-50 to-blue-50">
<tr>
<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">שם קובץ</th>
<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">מעלה</th>
<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">קטגוריה</th>
<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">גודל</th>
<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">תאריך העלאה</th>
<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">הורדות</th>
<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">סטטוס</th>
<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">פעולות</th>
```

```
</tr>

</thead>

<tbody class="divide-y divide-gray-200">

@for (file of files(); track trackByFileId($index, file)) {

<tr class="hover:bg-gray-50">

<td class="px-6 py-4">

<div class="flex items-center">

<div class="p-2 rounded-lg bg-gradient-to-br from-blue-100 to-blue-200 ml-3">

<i-lucide name="file-text" class="h-5 w-5 text-blue-600"></i-lucide>

</div>

<div>

<div class="text-sm font-medium text-gray-900">{{ file.originalName }}</div>

<div class="text-xs text-gray-500">{{ file.mimeType }}</div>

</div>

</div>

</td>

<td class="px-6 py-4 text-sm text-gray-900">{{ file.uploaderName }}</td>

<td class="px-6 py-4">

<span class="inline-block px-2 py-1 text-xs rounded-full bg-purple-100 text-purple-800">

{{ file.categoryName }}

</span>

</td>

<td class="px-6 py-4 text-sm text-gray-900">{{ formatFileSize(file.fileSize) }}</td>

<td class="px-6 py-4 text-sm text-gray-900">{{ formatDate(file.uploadedAt) }}</td>

<td class="px-6 py-4 text-sm text-gray-900">{{ file.downloadCount }}</td>
```

```
<td class="px-6 py-4">
<span
class="inline-block px-2 py-1 text-xs rounded-full"
[ngClass]="{
'bg-yellow-100 text-yellow-800': file.status === 'pending',
'bg-green-100 text-green-800': file.status === 'approved',
'bg-red-100 text-red-800': file.status === 'rejected'
}"
>
{{ getStatusText(file.status) }}
</span>
</td>
<td class="px-6 py-4">
<div class="flex items-center space-x-2 space-x-reverse">
@if (file.status === 'pending') {
<button
(click)="approveFile(file.id)"
class="p-2 bg-gradient-to-r from-green-500 to-emerald-500 hover:from-green-600 hover:to-emerald-600 text-white rounded-lg transition-all duration-200"
title="אשר קובץ"
>
<i-lucide name="check-circle" class="h-4 w-4"></i-lucide>
</button>
<button
(click)="rejectFile(file.id)"
class="p-2 bg-gradient-to-r from-red-500 to-pink-500 hover:from-red-600 hover:to-pink-600 text-white rounded-lg transition-all duration-200"
title="דחה קובץ"
>
<i-lucide name="x-circle" class="h-4 w-4"></i-lucide>
</button>
}
<button
(click)="downloadFile(file.id, file.originalName)"
class="p-2 border border-gray-300 rounded-lg hover:bg-blue-50 hover:border-blue-300 transition-all duration-200"
title="הורד קובץ"
>
<i-lucide name="download" class="h-4 w-4"></i-lucide>
</button>
<button
(click)="deleteFile(file.id)"
class="p-2 border border-gray-300 rounded-lg hover:bg-red-50 hover:border-red-300 text-red-600 transition-all duration-200"
title="מחק קובץ"
>
<i-lucide name="trash-2" class="h-4 w-4"></i-lucide>
</button>
</div>
</td>
</tr>
} @empty {
```

```
<tr>
<td colSpan="8" class="px-6 py-8 text-center text-gray-500">
<i-lucide name="file-text" class="mx-auto h-12 w-12 mb-4 text-gray-300"></i-lucide>
<p>אין קבצים להצגה</p>
<p class="text-sm">העלה קובץ חדש או שנה את הפילטרים</p>
</td>
</tr>
}
</tbody>
</table>
</div>
<!-- Pagination -->
@if (pagination()) {
<div class="bg-gray-50 px-6 py-3 flex items-center justify-between border-t border-gray-200">
<div class="text-sm text-gray-700">
מציג {{ (pagination()!.page - 1) * pagination()!.limit + 1 }} -
<!-- {{ Math.min(pagination()!.page * pagination()!.limit, pagination()!.total) }} -->
מתוך {{ pagination()!.total }}  קבצים
</div>
<div class="flex space-x-2 space-x-reverse">
<button
(click)="previousPage()"
[disabled]="pagination()!.page <= 1"
class="px-3 py-1 border border-gray-300 rounded disabled:opacity-50 disabled:cursor-not-allowed hover:bg-gray-100"
```

```
>
הקודם
</button>
<span class="px-3 py-1 bg-purple-100 text-purple-800 rounded">
{{ pagination()!.page }} / {{ pagination()!.totalPages }}
</span>
<button
(click)="nextPage()"
[disabled]="pagination()!.page >= pagination()!.totalPages"
class="px-3 py-1 border border-gray-300 rounded disabled:opacity-50 disabled:cursor-not-allowed hover:bg-gray-100"
>
הבא
</button>
</div>
</div>
}
</div>
<!-- Loading State -->
@if (apiService.loading()) {
<div class="text-center py-8">
<div class="inline-flex items-center px-4 py-2 font-semibold leading-6 text-sm shadow rounded-md text-white bg-purple-500">
<svg class="animate-spin -ml-1 mr-3 h-5 w-5 text-white" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24">
<circle class="opacity-25" cx="12" cy="12" r="10" stroke="currentColor" stroke-width="4"></circle>
```

```
<path class="opacity-75" fill="currentColor" d="M4 12a8 8 0 018-8V0C5.373 0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0 014 12H0c0 3.042 1.135 5.824 3 7.938l3-2.647z"></path>

</svg>

טוען קבצים...

</div>

</div>

}

<!-- Error State -->

@if (apiService.error()) {

<div class="bg-red-50 border border-red-200 rounded-lg p-4">

<div class="flex items-center">

<i-lucide name="alert-circle" class="h-5 w-5 text-red-500 ml-2"></i-lucide>

<span class="text-red-700">{{ apiService.error() }}</span>

</div>

</div>

}

</div>

import { Component, signal } from '@angular/core';

import { CommonModule } from '@angular/common';

import { FormsModule } from '@angular/forms';

import { LucideAngularModule } from 'lucide-angular';

@Component({

selector: 'app-header',

standalone: true,

imports: [CommonModule, FormsModule, LucideAngularModule],
templateUrl: './header.component.html',

styleUrls: ['./header.component.css']

})

export class HeaderComponent {

searchTerm = signal(");

}

<header class="bg-white/80 backdrop-blur-sm shadow-sm border-b border-gray-200 p-4">

<div class="flex items-center justify-between">

<div class="flex items-center space-x-4 space-x-reverse">

<h1 class="text-2xl font-bold bg-gradient-to-r from-purple-600 to-blue-600 bg-clip-text text-transparent">

Angular 19 Dashboard

</h1>

<span class="px-3 py-1 bg-gradient-to-r from-green-500 to-emerald-500 text-white text-xs rounded-full flex items-center">

<div class="w-2 h-2 bg-white rounded-full ml-1 animate-pulse"></div>

Standalone Ready

</span>

</div>

<div class="flex items-center space-x-4 space-x-reverse">

<div class="relative">

<i-lucide name="search" class="absolute right-3 top-1/2 transform -translate-y-1/2 text-gray-400 h-4 w-4"></i-lucide>

<input

type="text"

placeholder="חיפוש..."
```

```
class="pr-10 w-64 px-3 py-2 border border-gray-300 rounded-lg focus:border-purple-500 focus:ring-1 focus:ring-purple-500 outline-none transition-colors"
[(ngModel)]="searchTerm"
/>
</div>
<button class="p-2 border border-gray-300 rounded-lg hover:bg-gradient-to-r hover:from-purple-50 hover:to-blue-50 transition-all duration-200">
<i-lucide name="bell" class="h-4 w-4"></i-lucide>
</button>
<div class="w-8 h-8 bg-gradient-to-br from-purple-400 to-blue-400 rounded-full flex items-center justify-center text-white font-medium text-sm ring-2 ring-purple-200">
 מנ
</div>
</div>
</div>
</header>
import { Component, signal, inject, OnInit, OnDestroy } from '@angular/core';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { LucideAngularModule } from 'lucide-angular';
import { Subject, takeUntil } from 'rxjs';
import { ApiService } from '../../services/api.service.service';
interface SystemSettings {
siteName: string;
siteDescription: string;
maxFileSize: number;
allowedFileTypes: string[];
```

```
autoApproveFiles: boolean;
emailNotifications: boolean;
maintenanceMode: boolean;
backupFrequency: string;
}
@Component({
selector: 'app-settings',
standalone: true,
imports: [CommonModule, FormsModule, LucideAngularModule],
templateUrl: './settings.component.html',
styleUrls: ['./settings.component.css']
})
export class SettingsComponent implements OnInit, OnDestroy {
apiService = inject(ApiService);
private destroy$ = new Subject<void>();
settings = signal<SystemSettings>({
siteName: 'TEACH SHARE',
siteDescription: 'פלטפורמה לשיתוף חומרי לימוד',
maxFileSize: 50,
allowedFileTypes: ['pdf', 'doc', 'docx', 'ppt', 'pptx', 'xls', 'xlsx'],
autoApproveFiles: false,
emailNotifications: true,
maintenanceMode: false,
backupFrequency: 'daily'
});
```

```
passwordForm = {
currentPassword: ",
newPassword: ",
confirmPassword: "
};
showPasswordModal = signal(false);
showBackupModal = signal(false);
ngOnInit() {
this.loadSettings();
}
ngOnDestroy() {
this.destroy$.next();
this.destroy$.complete();
}
loadSettings() {
// This would load from your API
// For now, we'll use the default values
console.log('Loading settings...');
}
saveGeneralSettings() {
// Save general settings via API
console.log('Saving general settings:', this.settings());
// Simulate API call
setTimeout(() => {
alert('הגדרות נשמרו בהצלחה!');
}, 500);
}
openPasswordModal() {
this.passwordForm = {
currentPassword: ",
newPassword: ",
confirmPassword: "
};
this.showPasswordModal.set(true);
}
closePasswordModal() {
this.showPasswordModal.set(false);
}
changePassword() {
if (this.passwordForm.newPassword !== this.passwordForm.confirmPassword) {
alert('הסיסמאות אינן תואמות');
return;
}
if (this.passwordForm.newPassword.length < 6) {
alert('הסיסמה חייבת להכיל לפחות 6 תווים');
return;
}
// Change password via API
console.log('Changing password...');
// Simulate API call
```

```
setTimeout(() => {
alert('הסיסמה שונתה בהצלחה!');
this.closePasswordModal();
}, 500);
}
openBackupModal() {
this.showBackupModal.set(true);
}
closeBackupModal() {
this.showBackupModal.set(false);
}
createBackup() {
console.log('Creating backup...');
// Simulate backup creation
setTimeout(() => {
alert('גיבוי נוצר בהצלחה!');
this.closeBackupModal();
}, 2000);
}
exportData() {
console.log('Exporting data...');
// Simulate data export
const data = {
users: 'users_data',
files: 'files_data',
categories: 'categories_data',
exportDate: new Date().toISOString()
};
const blob = new Blob([JSON.stringify(data, null, 2)], { type: 'application/json' });
const url = window.URL.createObjectURL(blob);
const link = document.createElement('a');
link.href = url;
link.download = `teach-share-export-${new Date().toISOString().split('T')[0]}.json`;
link.click();
window.URL.revokeObjectURL(url);
}
clearCache() {
if (!confirm('האם אתה בטוח שברצונך לנקות את המטמון? פעולה זו עלולה להאט את המערכת זמנית.')) {
return;
}
console.log('Clearing cache...');
// Simulate cache clearing
setTimeout(() => {
alert('המטמון נוקה בהצלחה!');
}, 1000);
}
toggleMaintenanceMode() {
const currentSettings = this.settings();
const newMode = !currentSettings.maintenanceMode;
if (newMode) {
```

```
if (!confirm('האם אתה בטוח שברצונך להפעיל מצב תחזוקה? המשתמשים לא יוכלו לגשת לאתר.')) {
return;
}
}
this.settings.set({
...currentSettings,
maintenanceMode: newMode
});
console.log('Maintenance mode:', newMode ? 'enabled' : 'disabled');
}
updateSettings(field: keyof SystemSettings, value: any) {
const currentSettings = this.settings();
this.settings.set({
...currentSettings,
[field]: value
});
}
getSystemStatus() {
return {
serverStatus: 'פעיל',
databaseStatus: 'פעיל',
storageStatus: 'פעיל',
lastBackup: 'לפני 2 שעות',
uptime: '3 ימים, 15 שעות'
};
```

```
}
}
<!-- <div class="space-y-6">
<h2 class="text-xl font-bold">הגדרות מערכת</h2>
<div class="grid grid-cols-1 lg:grid-cols-2 gap-6">
General Settings -->
<!-- <div class="border-0 shadow-lg bg-white/80 backdrop-blur-sm rounded-lg p-6">
<h3 class="text-lg font-semibold mb-4 flex items-center">
<i-lucide name="settings" class="ml-2 h-5 w-5 text-purple-600"></i-lucide>
 הגדרות כלליות
</h3>
<div class="space-y-4">
<div>
<label class="block text-sm font-medium text-gray-700 mb-2">שם האתר</label>
<input
type="text"
[value]="settings().siteName"
(input)="updateSettings('siteName', $event.target.value)"
class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-purple-500 focus:ring-1 focus:ring-purple-500"
/>
</div>
<div>
<label class="block text-sm font-medium text-gray-700 mb-2">תיאור האתר</label>
<textarea
[value]="settings().siteDescription"
```

```
(input)="updateSettings('siteDescription', $event.target.value)"
rows="3"
class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-purple-500 focus:ring-1 focus:ring-purple-500"
></textarea>
</div>
<div>
<label class="block text-sm font-medium text-gray-700 mb-2">גודל קובץ מקסימלי (MB)</label>
<input
type="number"
[value]="settings().maxFileSize"
(input)="updateSettings('maxFileSize', +$event.target.value)"
min="1"
max="100"
class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-purple-500 focus:ring-1 focus:ring-purple-500"
/>
</div>
<div class="space-y-2">
<div class="flex items-center">
<input
type="checkbox"
[checked]="settings().autoApproveFiles"
(change)="updateSettings('autoApproveFiles', $event.target.checked)"
id="autoApprove"
class="ml-2"
```

```
/>
<label for="autoApprove" class="text-sm text-gray-700"> אישור אוטומטי לקבצים</label>
</div>
<div class="flex items-center">
<input
type="checkbox"
[checked]="settings().emailNotifications"
(change)="updateSettings('emailNotifications', $event.target.checked)"
id="emailNotifications"
class="ml-2"
/>
<label for="emailNotifications" class="text-sm text-gray-700">התראות אימייל</label>
</div>
</div>
<button
(click)="saveGeneralSettings()"
class="w-full px-4 py-2 bg-gradient-to-r from-purple-500 to-blue-500 hover:from-purple-600 hover:to-blue-600 text-white rounded-lg"
>
 שמור הגדרות
</button>
</div>
</div>
<!-- Security Settings -->
<div class="border-0 shadow-lg bg-white/80 backdrop-blur-sm rounded-lg p-6">
```

```
<h3 class="text-lg font-semibold mb-4 flex items-center">
<i-lucide name="shield" class="ml-2 h-5 w-5 text-red-600"></i-lucide>
 הגדרות אבטחה
</h3>
<div class="space-y-4">
<button
(click)="openPasswordModal()"
class="w-full px-4 py-2 bg-gradient-to-r from-red-500 to-pink-500 hover:from-red-600 hover:to-pink-600 text-white rounded-lg flex items-center justify-center"
>
<i-lucide name="key" class="w-4 h-4 ml-2"></i-lucide>
 שנה סיסמה
</button>
<div class="border-t pt-4">
<div class="flex items-center justify-between">
<span class="text-sm font-medium text-gray-700">מצב תחזוקה</span>
<button
(click)="toggleMaintenanceMode()"
class="px-3 py-1 rounded-full text-xs font-medium transition-colors"
[ngClass]="{
'bg-red-100 text-red-800': settings().maintenanceMode,
'bg-green-100 text-green-800': !settings().maintenanceMode
}"
>
{{ settings().maintenanceMode ? 'פעיל' : 'כבוי' }}
</button>
```

```
</div>

<p class="text-xs text-gray-500 mt-1">

במצב תחזוקה המשתמשים לא יוכלו לגשת לאתר

</p>

</div>

</div>

</div>

<!-- System Status -->

<div class="border-0 shadow-lg bg-white/80 backdrop-blur-sm rounded-lg p-6">

<h3 class="text-lg font-semibold mb-4 flex items-center">

<i-lucide name="activity" class="ml-2 h-5 w-5 text-green-600"></i-lucide>

סטטוס מערכת

</h3>

<div class="space-y-3">

<div class="flex items-center justify-between">

<span class="text-sm">שרת</span>

<span class="flex items-center text-green-600">

<div class="w-2 h-2 bg-green-500 rounded-full ml-2"></div>

{{ getSystemStatus().serverStatus }}

</span>

</div>

<div class="flex items-center justify-between">

<span class="text-sm">מסד נתונים</span>

<span class="flex items-center text-green-600">

<div class="w-2 h-2 bg-green-500 rounded-full ml-2"></div>
```

```
{{ getSystemStatus().databaseStatus }}
</span>
</div>
<div class="flex items-center justify-between">
<span class="text-sm">אחסון</span>
<span class="flex items-center text-green-600">
<div class="w-2 h-2 bg-green-500 rounded-full ml-2"></div>
{{ getSystemStatus().storageStatus }}
</span>
</div>
<div class="flex items-center justify-between">
<span class="text-sm">זמן פעילות</span>
<span class="text-sm text-gray-600">{{ getSystemStatus().uptime }}</span>
</div>
<div class="flex items-center justify-between">
<span class="text-sm">גיבוי אחרון</span>
<span class="text-sm text-gray-600">{{ getSystemStatus().lastBackup }}</span>
</div>
</div>
</div> -->
<!-- Backup & Maintenance -->
<!-- <div class="border-0 shadow-lg bg-white/80 backdrop-blur-sm rounded-lg p-6">
<h3 class="text-lg font-semibold mb-4 flex items-center">
<i-lucide name="database" class="ml-2 h-5 w-5 text-blue-600"></i-lucide>
 גיבוי ותחזוקה
```

```
</h3>
<div class="space-y-3">
<button
(click)="openBackupModal()"
class="w-full px-4 py-2 bg-gradient-to-r from-blue-500 to-cyan-500 hover:from-blue-600 hover:to-cyan-600 text-white rounded-lg flex items-center justify-center"
>
<i-lucide name="download" class="w-4 h-4 ml-2"></i-lucide>
 צור גיבוי
</button>
<button
(click)="exportData()"
class="w-full px-4 py-2 border border-gray-300 rounded-lg hover:bg-gray-50 flex items-center justify-center"
>
<i-lucide name="file-down" class="w-4 h-4 ml-2"></i-lucide>
 ייצא נתונים
</button>
<button
(click)="clearCache()"
class="w-full px-4 py-2 border border-gray-300 rounded-lg hover:bg-gray-50 flex items-center justify-center"
>
<i-lucide name="trash-2" class="w-4 h-4 ml-2"></i-lucide>
 נקה מטמון
</button>
<div class="border-t pt-3">
```

```
<label class="block text-sm font-medium text-gray-700 mb-2">תדירות גיבוי</label>
<select
[value]="settings().backupFrequency"
(change)="updateSettings('backupFrequency', $event.target.value)"
class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-blue-500 focus:ring-1 focus:ring-blue-500"
>
<option value="hourly">כל שעה</option>
<option value="daily">יומי</option>
<option value="weekly">שבועי</option>
<option value="monthly">חודשי</option>
</select>
</div>
</div>
</div>
</div>
<!-- Password Change Modal -->
@if (showPasswordModal()) {
<div class="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50">
<div class="bg-white rounded-lg p-6 w-full max-w-md mx-4">
<div class="flex justify-between items-center mb-4">
<h3 class="text-lg font-semibold">שינוי סיסמה</h3>
<button (click)="closePasswordModal()" class="text-gray-400 hover:text-gray-600">
<i-lucide name="x" class="h-5 w-5"></i-lucide>
</button>
```

```
</div>
<form (ngSubmit)="changePassword()" #passwordFormRef="ngForm">
<div class="space-y-4">
<div>
<label class="block text-sm font-medium text-gray-700 mb-1">סיסמה נוכחית</label>
<input
type="password"
[(ngModel)]="passwordForm.currentPassword"
name="currentPassword"
required
class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-red-500 focus:ring-1 focus:ring-red-500"
/>
</div>
<div>
<label class="block text-sm font-medium text-gray-700 mb-1">סיסמה חדשה</label>
<input
type="password"
[(ngModel)]="passwordForm.newPassword"
name="newPassword"
required
minlength="6"
class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-red-500 focus:ring-1 focus:ring-red-500"
/>
</div>
```

```
<div>
<label class="block text-sm font-medium text-gray-700 mb-1">אימות סיסמה</label>
<input
type="password"
[(ngModel)]="passwordForm.confirmPassword"
name="confirmPassword"
required
class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-red-500 focus:ring-1 focus:ring-red-500"
/>
</div>
</div>
<div class="flex justify-end space-x-2 space-x-reverse mt-6">
<button
type="button"
(click)="closePasswordModal()"
class="px-4 py-2 border border-gray-300 rounded-lg hover:bg-gray-50"
>
 ביטול
</button>
<button
type="submit"
[disabled]="!passwordFormRef.valid"
class="px-4 py-2 bg-gradient-to-r from-red-500 to-pink-500 hover:from-red-600 hover:to-pink-600 text-white rounded-lg disabled:opacity-50"
>
```

```
שנה סיסמה
</button>
</div>
</form>
</div>
</div>
} -->
<!-- Backup Modal -->
<!-- @if (showBackupModal()) {
<div class="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50">
<div class="bg-white rounded-lg p-6 w-full max-w-md mx-4">
<div class="flex justify-between items-center mb-4">
<h3 class="text-lg font-semibold">יצירת גיבוי</h3>
<button (click)="closeBackupModal()" class="text-gray-400 hover:text-gray-600">
<i-lucide name="x" class="h-5 w-5"></i-lucide>
</button>
</div>
<div class="space-y-4">
<p class="text-sm text-gray-600">
הגיבוי יכלול את כל הנתונים: משתמשים, קבצים, קטגוריות והגדרות המערכת.
</p>
<div class="bg-yellow-50 border border-yellow-200 rounded-lg p-3">
<div class="flex items-center">
<i-lucide name="alert-triangle" class="h-4 w-4 text-yellow-600 ml-2"></i-lucide>
<span class="text-sm text-yellow-800">
```

```
תהליך הגיבוי עלול לקחת מספר דקות
</span>
</div>
</div>
<div class="flex justify-end space-x-2 space-x-reverse">
<button
(click)="closeBackupModal()"
class="px-4 py-2 border border-gray-300 rounded-lg hover:bg-gray-50"
>
ביטול
</button>
<button
(click)="createBackup()"
class="px-4 py-2 bg-gradient-to-r from-blue-500 to-cyan-500 hover:from-blue-600 hover:to-cyan-600 text-white rounded-lg"
>
התחל גיבוי
</button>
</div>
</div>
</div>
</div>
}
</div> -->
import { Component, signal } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterModule } from '@angular/router';
import { LucideAngularModule } from 'lucide-angular';
interface MenuItem {
id: string;
label: string;
icon: string;
route: string;
hoverClass: string;
}
@Component({
selector: 'app-sidebar',
standalone: true,
imports: [CommonModule, RouterModule, LucideAngularModule],
templateUrl: './sidebar.component.html',
styleUrls: ['./sidebar.component.css']
})
export class SidebarComponent {
menuItems = signal<MenuItem[]>([
{
id: 'dashboard',
label: 'דשבורד',
icon: 'bar-chart-3',
route: '/dashboard',
hoverClass: 'hover:from-purple-50 hover:to-blue-50 hover:text-purple-700'
},
{
id: 'users',
label: 'ניהול משתמשים',
icon: 'users',
route: '/users',
hoverClass: 'hover:from-blue-50 hover:to-cyan-50 hover:text-blue-700'
},
{
id: 'files',
label: 'ניהול קבצים',
icon: 'file-text',
route: '/files',
hoverClass: 'hover:from-green-50 hover:to-emerald-50 hover:text-green-700'
},
{
id: 'categories',
label: 'קטגוריות',
icon: 'folder-open',
route: '/categories',
hoverClass: 'hover:from-orange-50 hover:to-yellow-50 hover:text-orange-700'
},
{
id: 'settings',
label: 'הגדרות',
icon: 'settings',
route: '/settings',
hoverClass: 'hover:from-pink-50 hover:to-rose-50 hover:text-pink-700'
}
]);
getButtonClass(item: MenuItem): string {
return `text-gray-700 ${item.hoverClass}`;
}
}
<div class="w-64 bg-white shadow-xl border-l border-gray-200 relative overflow-hidden">
<!-- Gradient Background -->
<div class="absolute inset-0 bg-gradient-to-b from-purple-600 via-blue-600 to-cyan-600 opacity-5"></div>
<div class="relative z-10 p-6 border-b border-gray-200">
<div class="flex items-center space-x-3 space-x-reverse">
<div class="w-10 h-10 rounded-xl bg-gradient-to-br from-purple-500 via-blue-500 to-cyan-500 flex items-center justify-center shadow-lg">
<i-lucide name="star" class="w-6 h-6 text-white"></i-lucide>
</div>
<div>
<h2 class="text-xl font-bold bg-gradient-to-r from-purple-600 to-blue-600 bg-clip-text text-transparent">
TEACH SHARE
</h2>
<p class="text-sm text-gray-500">Angular 19 Admin</p>
</div>
</div>
```

```
</div>

<nav class="mt-6 relative z-10">

<div class="px-4 space-y-2">

@for (item of menuItems(); track item.id) {

<a

[routerLink]="item.route"

routerLinkActive="active-menu-item"

[class]="getButtonClass(item)"

class="w-full flex items-center px-4 py-3 text-right rounded-lg transition-all duration-200 group"

>

<i-lucide [name]="item.icon" class="ml-2 h-4 w-4"></i-lucide>

<span>{{ item.label }}</span>

</a>

}

</div>

</nav>

</div>

import { Component, signal, inject, OnInit, OnDestroy } from '@angular/core';

import { CommonModule } from '@angular/common';

import { FormsModule } from '@angular/forms';

import { LucideAngularModule } from 'lucide-angular';

import { Subject, takeUntil } from 'rxjs';

import { ApiService } from '../../services/api.service.service';

import { User, ApiResponse } from '../../models/api.models';

@Component({
```

```
selector: 'app-users',
standalone: true,
imports: [CommonModule, FormsModule, LucideAngularModule],
templateUrl: './users.component.html',
styleUrls: ['./users.component.css']
})
export class UsersComponent implements OnInit, OnDestroy {
apiService = inject(ApiService);
private destroy$ = new Subject<void>();
users = signal<User[]>([]);
pagination = signal<any>(null);
showUserModal = signal(false);
editingUser = signal<User | null>(null);
searchTerm = '';
currentPage = 1;
pageSize = 10;
searchTimeout: any;
userFormData: Partial<User> = {
firstName: '',
lastName: '',
email: '',
role: 'student',
isActive: true
};
ngOnInit() {
```

```
this.loadUsers();
}
ngOnDestroy() {
this.destroy$.next();
this.destroy$.complete();
if (this.searchTimeout) {
clearTimeout(this.searchTimeout);
}
}
loadUsers() {
this.apiService.getUsers(this.currentPage, this.pageSize, this.searchTerm)
.pipe(takeUntil(this.destroy$))
.subscribe({
next: (response) => {
this.users.set(response.data);
this.pagination.set(response.pagination);
},
error: (error) => {
console.error('Error loading users:', error);
}
});
}
onSearchChange() {
if (this.searchTimeout) {
clearTimeout(this.searchTimeout);
```

```
}
this.searchTimeout = setTimeout(() => {
this.currentPage = 1;
this.loadUsers();
}, 500);
}
openCreateUserModal() {
this.editingUser.set(null);
this.userFormData = {
firstName: '',
lastName: '',
email: '',
role: 'student',
isActive: true
};
this.showUserModal.set(true);
}
editUser(user: User) {
this.editingUser.set(user);
this.userFormData = {
firstName: user.firstName,
lastName: user.lastName,
email: user.email,
role: user.role,
isActive: user.isActive
```

```
};
this.showUserModal.set(true);
}
closeUserModal() {
this.showUserModal.set(false);
this.editingUser.set(null);
}
saveUser() {
const editingUser = this.editingUser();
if (editingUser) {
this.apiService.updateUser(editingUser.id, this.userFormData)
.pipe(takeUntil(this.destroy$))
.subscribe({
next: () => {
this.closeUserModal();
this.loadUsers();
},
error: (error) => {
console.error('Error updating user:', error);
}
});
} else {
this.apiService.createUser(this.userFormData)
.pipe(takeUntil(this.destroy$))
.subscribe({
```

```
next: () => {

this.closeUserModal();

this.loadUsers();

},

error: (error) => {

console.error('Error creating user:', error);

}

});

}

}

toggleUserStatus(user: User) {

const newStatus = !user.isActive;

const action = newStatus ? 'השבתת' : 'הפעלת';

if (!confirm(`האם אתה בטוח ב${action} המשתמש ${user.firstName}
${user.lastName}?`)) {

return;

}

this.apiService.updateUser(user.id, { isActive: newStatus })

.pipe(takeUntil(this.destroy$))

.subscribe({

next: () => {

this.loadUsers();

},

error: (error) => {

console.error('Error updating user status:', error);

}
```

```
});
}
deleteUser(userId: number) {
if (!confirm('האם אתה בטוח שברצונך למחוק את המשתמש? פעולה זו לא ניתנת לביטול.')) {
return;
}
this.apiService.deleteUser(userId)
.pipe(takeUntil(this.destroy$))
.subscribe({
next: () => {
this.loadUsers();
},
error: (error) => {
console.error('Error deleting user:', error);
}
});
}
previousPage() {
if (this.currentPage > 1) {
this.currentPage--;
this.loadUsers();
}
}
nextPage() {
const pagination = this.pagination();
```

```
if (pagination && this.currentPage < pagination.totalPages) {

this.currentPage++;

this.loadUsers();

}

}

trackByUserId(index: number, user: User): number {

return user.id;

}

getUserInitials(user: User): string {

return (user.firstName.charAt(0) + user.lastName.charAt(0)).toUpperCase();

}

getRoleText(role: string): string {

switch (role) {

case 'admin': return 'מנהל';

case 'teacher': return 'מורה';

case 'student': return 'תלמיד';

default: return role;

}

}

formatDate(dateString: string): string {

const date = new Date(dateString);

return date.toLocaleDateString('he-IL');

}

}

<div class="space-y-6">
```

```
<!-- Header -->
<div class="flex justify-between items-center">
<h2 class="text-xl font-bold">ניהול משתמשים</h2>
<div class="flex space-x-2 space-x-reverse">
<div class="relative">
<i-lucide name="search" class="absolute right-3 top-1/2 transform -translate-y-1/2 text-gray-400 h-4 w-4"></i-lucide>
<input
type="text"
placeholder="חיפוש משתמש..."
class="pr-10 w-64 px-3 py-2 border border-gray-300 rounded-lg focus:border-purple-500 focus:ring-1 focus:ring-purple-500"
[(ngModel)]="searchTerm"
(input)="onSearchChange()"
/>
</div>
<button
(click)="loadUsers()"
class="px-4 py-2 border border-gray-300 rounded-lg hover:bg-gray-50 flex items-center"
>
<i-lucide name="refresh-cw" class="w-4 h-4 ml-2"></i-lucide>
 רענן
</button>
<button
(click)="openCreateUserModal()"
```

```
class="px-4 py-2 bg-gradient-to-r from-green-500 to-emerald-500 hover:from-green-600 hover:to-emerald-600 text-white rounded-lg flex items-center"

>

<i-lucide name="plus" class="w-4 h-4 ml-2"></i-lucide>

 הוסף משתמש

</button>

</div>

</div>

<!-- Users Table -->

<div class="border-0 shadow-lg bg-white/80 backdrop-blur-sm rounded-lg overflow-hidden">

<div class="overflow-x-auto">

<table class="w-full">

<thead class="bg-gradient-to-r from-green-50 to-emerald-50">

<tr>

<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">משתמש</th>

<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">אימייל</th>

<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">תפקיד</th>

<th class="px-6 py-3 text-right text-sm font-medium text-gray-700"> תאריך הצטרפות</th>

<th class="px-6 py-3 text-right text-sm font-medium text-gray-700"> התחברות אחרונה</th>

<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">סטטוס</th>

<th class="px-6 py-3 text-right text-sm font-medium text-gray-700">פעולות</th>

</tr>

</thead>

<tbody class="divide-y divide-gray-200">
```

```
@for (user of users(); track trackByUserId($index, user)) {
<tr class="hover:bg-gray-50">
<td class="px-6 py-4">
<div class="flex items-center">
<div class="w-10 h-10 bg-gradient-to-br from-purple-400 to-blue-400 rounded-full flex items-center justify-center text-white font-medium text-sm ml-3">
{{ getUserInitials(user) }}
</div>
<div>
<div class="text-sm font-medium text-gray-900">{{ user.firstName }} {{ user.lastName }}</div>
<div class="text-xs text-gray-500">ID: {{ user.id }}</div>
</div>
</div>
</td>
<td class="px-6 py-4 text-sm text-gray-900">{{ user.email }}</td>
<td class="px-6 py-4">
<span
class="inline-block px-2 py-1 text-xs rounded-full"
[ngClass]="{
'bg-red-100 text-red-800': user.role === 'admin',
'bg-blue-100 text-blue-800': user.role === 'teacher',
'bg-gray-100 text-gray-800': user.role === 'student'
}"
>
{{ getRoleText(user.role) }}
```

```
</span>

</td>

<td class="px-6 py-4 text-sm text-gray-900">{{ formatDate(user.createdAt) }}</td>

<td class="px-6 py-4 text-sm text-gray-900">

{{ user.lastLogin ? formatDate(user.lastLogin) : 'מעולם לא' }}

</td>

<td class="px-6 py-4">

<span

class="inline-block px-2 py-1 text-xs rounded-full"

[ngClass]="{

'bg-green-100 text-green-800': user.isActive,

'bg-red-100 text-red-800': !user.isActive

}"

>

{{ user.isActive ? 'לא פעיל' : 'פעיל' }}

</span>

</td>

<td class="px-6 py-4">

<div class="flex items-center space-x-2 space-x-reverse">

<button

(click)="editUser(user)"

class="p-2 border border-gray-300 rounded-lg hover:bg-blue-50 hover:border-blue-300 transition-all duration-200"

title="ערוך משתמש"

>

<i-lucide name="edit" class="h-4 w-4"></i-lucide>
```

```
</button>
<button
(click)="toggleUserStatus(user)"
class="p-2 border border-gray-300 rounded-lg transition-all duration-200"
[ngClass]="{
'hover:bg-red-50 hover:border-red-300 text-red-600': user.isActive,
'hover:bg-green-50 hover:border-green-300 text-green-600': !user.isActive
}"
[title]="user.isActive ? 'השבת משתמש' : 'הפעל משתמש'"
>
<i-lucide [name]="user.isActive ? 'user-x' : 'user-check'" class="h-4 w-4"></i-lucide>
</button>
<button
(click)="deleteUser(user.id)"
class="p-2 border border-gray-300 rounded-lg hover:bg-red-50 hover:border-red-300 text-red-600 transition-all duration-200"
title="מחק משתמש"
>
<i-lucide name="trash-2" class="h-4 w-4"></i-lucide>
</button>
</div>
</td>
</tr>
} @empty {
<tr>
<td colSpan="7" class="px-6 py-8 text-center text-gray-500">
```

```
<i-lucide name="users" class="mx-auto h-12 w-12 mb-4 text-gray-300"></i-lucide>

<p>אין משתמשים להצגה</p>

<p class="text-sm">הוסף משתמש חדש או שנה את החיפוש</p>

</td>

</tr>

}

</tbody>

</table>

</div>

<!-- Pagination -->

@if (pagination()) {

<div class="bg-gray-50 px-6 py-3 flex items-center justify-between border-t border-gray-200">

<div class="text-sm text-gray-700">

מציג {{ (pagination()!.page - 1) * pagination()!.limit + 1 }} -

<!-- {{ Math.min(pagination()!.page * pagination()!.limit, pagination()!.total) }} -->

מתוך {{ pagination()!.total }}  משתמשים

</div>

<div class="flex space-x-2 space-x-reverse">

<button

(click)="previousPage()"

[disabled]="pagination()!.page <= 1"

class="px-3 py-1 border border-gray-300 rounded disabled:opacity-50 disabled:cursor-not-allowed hover:bg-gray-100"

>

 הקודם
```

```
</button>

<span class="px-3 py-1 bg-green-100 text-green-800 rounded">

{{ pagination()!.page }} / {{ pagination()!.totalPages }}

</span>

<button

(click)="nextPage()"

[disabled]="pagination()!.page >= pagination()!.totalPages"

class="px-3 py-1 border border-gray-300 rounded disabled:opacity-50 disabled:cursor-not-allowed hover:bg-gray-100"

>

 הבא

</button>

</div>

</div>

}

</div>

<!-- Create/Edit User Modal -->

@if (showUserModal()) {

<div class="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50">

<div class="bg-white rounded-lg p-6 w-full max-w-md mx-4">

<div class="flex justify-between items-center mb-4">

<h3 class="text-lg font-semibold">

{{ editingUser() ? 'ערוך משתמש' : 'הוסף משתמש חדש' }}

</h3>

<button (click)="closeUserModal()" class="text-gray-400 hover:text-gray-600">
```

```
<i-lucide name="x" class="h-5 w-5"></i-lucide>
</button>
</div>
<form (ngSubmit)="saveUser()" #userForm="ngForm">
<div class="space-y-4">
<div>
<label class="block text-sm font-medium text-gray-700 mb-1">שם פרטי</label>
<input
type="text"
[(ngModel)]="userFormData.firstName"
name="firstName"
required
class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-purple-500 focus:ring-1 focus:ring-purple-500"
/>
</div>
<div>
<label class="block text-sm font-medium text-gray-700 mb-1">שם משפחה</label>
<input
type="text"
[(ngModel)]="userFormData.lastName"
name="lastName"
required
class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-purple-500 focus:ring-1 focus:ring-purple-500"
/>
```

```
</div>
<div>
<label class="block text-sm font-medium text-gray-700 mb-1">אימייל</label>
<input
type="email"
[(ngModel)]="userFormData.email"
name="email"
required
class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-purple-500 focus:ring-1 focus:ring-purple-500"
/>
</div>
<div>
<label class="block text-sm font-medium text-gray-700 mb-1">תפקיד</label>
<select
[(ngModel)]="userFormData.role"
name="role"
required
class="w-full px-3 py-2 border border-gray-300 rounded-lg focus:border-purple-500 focus:ring-1 focus:ring-purple-500"
>
<option value="student">תלמיד</option>
<option value="teacher">מורה</option>
<option value="admin">מנהל</option>
</select>
</div>
```

```
<div class="flex items-center">
<input
type="checkbox"
[(ngModel)]="userFormData.isActive"
name="isActive"
id="isActive"
class="ml-2"
/>
<label for="isActive" class="text-sm text-gray-700">משתמש פעיל</label>
</div>
</div>
<div class="flex justify-end space-x-2 space-x-reverse mt-6">
<button
type="button"
(click)="closeUserModal()"
class="px-4 py-2 border border-gray-300 rounded-lg hover:bg-gray-50"
>
ביטול
</button>
<button
type="submit"
[disabled]="!userForm.valid || apiService.loading()"
class="px-4 py-2 bg-gradient-to-r from-green-500 to-emerald-500 hover:from-green-600 hover:to-emerald-600 text-white rounded-lg disabled:opacity-50"
>
{{ editingUser() ? 'הוסף' : 'עדכן' }}
```

```
</button>
</div>
</form>
</div>
</div>
}
<!-- Loading State -->
@if (apiService.loading()) {
<div class="text-center py-8">
<div class="inline-flex items-center px-4 py-2 font-semibold leading-6 text-sm shadow rounded-md text-white bg-green-500">
<svg class="animate-spin -ml-1 mr-3 h-5 w-5 text-white" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24">
<circle class="opacity-25" cx="12" cy="12" r="10" stroke="currentColor" stroke-width="4"></circle>
<path class="opacity-75" fill="currentColor" d="M4 12a8 8 0 018-8V0C5.373 0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0 014 12H0c0 3.042 1.135 5.824 3 7.938l3-2.647z"></path>
</svg>
טוען משתמשים...
</div>
</div>
}
<!-- Error State -->
@if (apiService.error()) {
<div class="bg-red-50 border border-red-200 rounded-lg p-4">
<div class="flex items-center">
<i-lucide name="alert-circle" class="h-5 w-5 text-red-500 ml-2"></i-lucide>
```

```
<span class="text-red-700">{{ apiService.error() }}</span>

</div>

</div>

}

</div>

export interface User {

id: number;

firstName: string;

lastName: string;

email: string;

role: 'admin' | 'teacher' | 'student';

isActive: boolean;

createdAt: string;

lastLogin?: string;

profileImage?: string;

}

export interface FileItem {

id: number;

fileName: string;

originalName: string;

fileSize: number;

mimeType: string;

uploadedBy: number;

uploaderName: string;

categoryId: number;
```

```
categoryName: string;
status: 'pending' | 'approved' | 'rejected';
downloadCount: number;
uploadedAt: string;
approvedAt?: string;
approvedBy?: number;
description?: string;
tags?: string[];
}
export interface Category {
id: number;
name: string;
description?: string;
parentId?: number;
isActive: boolean;
fileCount: number;
createdAt: string;
updatedAt: string;
}
export interface Stats {
totalUsers: number;
activeUsers: number;
totalFiles: number;
pendingFiles: number;
approvedFiles: number;
```

```
totalDownloads: number;
todayDownloads: number;
totalCategories: number;
storageUsed: number; // in bytes
storageLimit: number; // in bytes
}
export interface ApiResponse<T> {
success: boolean;
data: T;
message?: string;
pagination?: {
page: number;
limit: number;
total: number;
totalPages: number;
};
}
export interface LoginRequest {
email: string;
password: string;
}
export interface LoginResponse {
token: string;
refreshToken: string;
user: User;
```

```
expiresIn: number;
}
import { Injectable, inject, signal } from '@angular/core';
import { HttpClient, HttpParams, HttpHeaders } from '@angular/common/http';
import { Observable, BehaviorSubject, throwError } from 'rxjs';
import { catchError, map, tap } from 'rxjs/operators';
import {
User,
FileItem,
Category,
Stats,
ApiResponse,
LoginRequest,
LoginResponse
} from '../models/api.models';
@Injectable({
providedIn: 'root'
})
export class ApiService {
private http = inject(HttpClient);
private baseUrl = 'http://localhost:5103/api'; // שנה לכתובת השרת שלך
// State management with signals
loading = signal(false);
error = signal<string | null>(null);
// Auth state
```

```
private tokenSubject = new BehaviorSubject<string | null>(
localStorage.getItem('token')
);
public token$ = this.tokenSubject.asObservable();
constructor() {
// Set auth header if token exists
const token = localStorage.getItem('token');
if (token) {
this.setAuthHeader(token);
}
}
private setAuthHeader(token: string) {
this.tokenSubject.next(token);
}
private getAuthHeaders(): HttpHeaders {
const token = this.tokenSubject.value;
return new HttpHeaders({
'Content-Type': 'application/json',
...(token && { 'Authorization': `Bearer ${token}` })
});
}
private handleError(error: any) {
console.error('API Error:', error);
this.loading.set(false);
let errorMessage = 'שגיאה לא ידועה';
```

```
if (error.error?.message) {
errorMessage = error.error.message;
} else if (error.message) {
errorMessage = error.message;
} else if (error.status === 0) {
errorMessage = 'שגיאת חיבור לשרת';
} else if (error.status === 401) {
errorMessage = 'אין הרשאה - נא להתחבר מחדש';
this.logout();
} else if (error.status === 403) {
errorMessage = 'אין הרשאה לפעולה זו';
} else if (error.status === 404) {
errorMessage = 'המשאב לא נמצא';
} else if (error.status >= 500) {
errorMessage = 'שגיאת שרת פנימית';
}
this.error.set(errorMessage);
return throwError(() => new Error(errorMessage));
}
// Auth Methods
login(credentials: LoginRequest): Observable<LoginResponse> {
this.loading.set(true);
this.error.set(null);
return this.http.post<ApiResponse<LoginResponse>>(
`${this.baseUrl}/auth/login`,
credentials
).pipe(
map(response => response.data),
tap(loginData => {
localStorage.setItem('token', loginData.token);
localStorage.setItem('refreshToken', loginData.refreshToken);
localStorage.setItem('user', JSON.stringify(loginData.user));
this.setAuthHeader(loginData.token);
this.loading.set(false);
}),
catchError(this.handleError.bind(this))
);
}
logout(): void {
localStorage.removeItem('token');
localStorage.removeItem('refreshToken');
localStorage.removeItem('user');
this.tokenSubject.next(null);
}
// Stats Methods
getStats(): Observable<Stats> {
this.loading.set(true);
this.error.set(null);
return this.http.get<ApiResponse<Stats>>(
`${this.baseUrl}/stats`,
{ headers: this.getAuthHeaders() }
).pipe(
map(response => response.data),
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
}
// Users Methods
getUsers(page: number = 1, limit: number = 10, search?: string): Observable<ApiResponse<User[]>> {
this.loading.set(true);
this.error.set(null);
let params = new HttpParams()
.set('page', page.toString())
.set('limit', limit.toString());
if (search) {
params = params.set('search', search);
}
return this.http.get<ApiResponse<User[]>>(
`${this.baseUrl}/users`,
{
headers: this.getAuthHeaders(),
params
}
).pipe(
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
}
getUserById(id: number): Observable<User> {
return this.http.get<ApiResponse<User>>(
`${this.baseUrl}/users/${id}`,
{ headers: this.getAuthHeaders() }
).pipe(
map(response => response.data),
catchError(this.handleError.bind(this))
);
}
createUser(user: Partial<User>): Observable<User> {
this.loading.set(true);
return this.http.post<ApiResponse<User>>(
`${this.baseUrl}/users`,
user,
{ headers: this.getAuthHeaders() }
).pipe(
map(response => response.data),
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
}
updateUser(id: number, user: Partial<User>): Observable<User> {
```

```
this.loading.set(true);
return this.http.put<ApiResponse<User>>(
`${this.baseUrl}/users/${id}`,
user,
{ headers: this.getAuthHeaders() }
).pipe(
map(response => response.data),
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
}
deleteUser(id: number): Observable<void> {
this.loading.set(true);
return this.http.delete<ApiResponse<void>>(
`${this.baseUrl}/users/${id}`,
{ headers: this.getAuthHeaders() }
).pipe(
map(() => void 0),
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
}
// Files Methods
getFiles(page: number = 1, limit: number = 10, status?: string, categoryId?: number): Observable<ApiResponse<FileItem[]>> {
this.loading.set(true);
this.error.set(null);
let params = new HttpParams()
.set('page', page.toString())
.set('limit', limit.toString());
if (status) {
params = params.set('status', status);
}
if (categoryId) {
params = params.set('categoryId', categoryId.toString());
}
return this.http.get<ApiResponse<FileItem[]>>(
`${this.baseUrl}/files`,
{
headers: this.getAuthHeaders(),
params
}
).pipe(
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
}
uploadFile(file: File, categoryId: number, description?: string): Observable<FileItem> {
this.loading.set(true);
const formData = new FormData();
formData.append('file', file);
formData.append('categoryId', categoryId.toString());
if (description) {
formData.append('description', description);
}
const headers = new HttpHeaders({
'Authorization': `Bearer ${this.tokenSubject.value}`
});
return this.http.post<ApiResponse<FileItem>>(
`${this.baseUrl}/files/upload`,
formData,
{ headers }
).pipe(
map(response => response.data),
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
}
approveFile(id: number): Observable<FileItem> {
this.loading.set(true);
return this.http.put<ApiResponse<FileItem>>(
`${this.baseUrl}/files/${id}/approve`,
{},
{ headers: this.getAuthHeaders() }
).pipe(
map(response => response.data),
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
}
rejectFile(id: number, reason?: string): Observable<void> {
this.loading.set(true);
return this.http.put<ApiResponse<void>>(
`${this.baseUrl}/files/${id}/reject`,
{ reason },
{ headers: this.getAuthHeaders() }
).pipe(
map(() => void 0),
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
}
deleteFile(id: number): Observable<void> {
this.loading.set(true);
return this.http.delete<ApiResponse<void>>(
`${this.baseUrl}/files/${id}`,
{ headers: this.getAuthHeaders() }
).pipe(
map(() => void 0),
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
```

```
);
}
downloadFile(id: number): Observable<Blob> {
return this.http.get(
`${this.baseUrl}/files/${id}/download`,
{
headers: this.getAuthHeaders(),
responseType: 'blob'
}
).pipe(
catchError(this.handleError.bind(this))
);
}
// Categories Methods
getCategories(): Observable<Category[]> {
this.loading.set(true);
this.error.set(null);
return this.http.get<ApiResponse<Category[]>>(
`${this.baseUrl}/categories`,
{ headers: this.getAuthHeaders() }
).pipe(
map(response => response.data),
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
```

```
}
createCategory(category: Partial<Category>): Observable<Category> {
this.loading.set(true);
return this.http.post<ApiResponse<Category>>(
`${this.baseUrl}/categories`,
category,
{ headers: this.getAuthHeaders() }
).pipe(
map(response => response.data),
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
}
updateCategory(id: number, category: Partial<Category>): Observable<Category> {
this.loading.set(true);
return this.http.put<ApiResponse<Category>>(
`${this.baseUrl}/categories/${id}`,
category,
{ headers: this.getAuthHeaders() }
).pipe(
map(response => response.data),
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
}
```

```
deleteCategory(id: number): Observable<void> {
this.loading.set(true);
return this.http.delete<ApiResponse<void>>(
`${this.baseUrl}/categories/${id}`,
{ headers: this.getAuthHeaders() }
).pipe(
map(() => void 0),
tap(() => this.loading.set(false)),
catchError(this.handleError.bind(this))
);
}
}
import { Component, signal } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterOutlet } from '@angular/router';
import { LucideAngularModule } from 'lucide-angular';
import { DashboardComponent } from './components/dashboard/dashboard.component';
import { FilesComponent } from './components/files/files.component';
import { UsersComponent } from './components/users/users.component';
import { CategoriesComponent } from './components/categories/categories.component';
import { SettingsComponent } from './components/settings/settings.component';
@Component({
selector: 'app-root',
standalone: true,
imports: [
CommonModule,
RouterOutlet,
LucideAngularModule,
DashboardComponent,
FilesComponent,
UsersComponent,
CategoriesComponent,
SettingsComponent
],
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent {
activeTab = signal('dashboard');
menuItems = signal([
{ id: 'dashboard', label: 'דשבורד', icon: 'bar-chart-3', color: 'from-purple-600 to-blue-600' },
{ id: 'files', label: 'ניהול קבצים', icon: 'file-text', color: 'from-green-500 to-emerald-500' },
{ id: 'users', label: 'ניהול משתמשים', icon: 'users', color: 'from-blue-500 to-cyan-500' },
{ id: 'categories', label: 'קטגוריות', icon: 'folder-open', color: 'from-orange-500 to-yellow-500' },
{ id: 'settings', label: 'הגדרות', icon: 'settings', color: 'from-pink-500 to-rose-500' }
]);
setActiveTab(tabId: string) {
this.activeTab.set(tabId);
```

```
}

getButtonClass(itemId: string): string {

const baseClass = 'w-full flex items-center px-4 py-3 text-right rounded-lg transition-all duration-200';

const item = this.menuItems().find(i => i.id === itemId);

if (this.activeTab() === itemId) {

return `${baseClass} bg-gradient-to-r ${item?.color} text-white shadow-lg`;

}

return `${baseClass} text-gray-700 hover:bg-gradient-to-r hover:from-gray-50 hover:to-gray-100`;

}

getPageTitle(): string {

const titles: { [key: string]: string } = {

'dashboard': 'דשבורד ראשי',

'files': 'ניהול קבצים',

'users': 'ניהול משתמשים',

'categories': 'ניהול קטגוריות',

'settings': 'הגדרות מערכת'

};

return titles[this.activeTab()] || 'ממשק ניהול';

}

}

<div class="min-h-screen bg-gradient-to-br from-purple-50 via-blue-50 to-cyan-50 flex" dir="rtl">

<!-- Sidebar -->

<div class="w-64 bg-white shadow-xl border-l border-gray-200 relative overflow-hidden">
```

```
<div class="absolute inset-0 bg-gradient-to-b from-purple-600 via-blue-600 to-cyan-600 opacity-5"></div>

<div class="relative z-10 p-6 border-b border-gray-200">

<div class="flex items-center space-x-3 space-x-reverse">

<div class="w-10 h-10 rounded-xl bg-gradient-to-br from-purple-500 via-blue-500 to-cyan-500 flex items-center justify-center shadow-lg">

<i-lucide name="star" class="w-6 h-6 text-white"></i-lucide>

</div>

<div>

<h2 class="text-xl font-bold bg-gradient-to-r from-purple-600 to-blue-600 bg-clip-text text-transparent">

TEACH SHARE

</h2>

<p class="text-sm text-gray-500">ממשק ניהול</p>

</div>

</div>

</div>

<nav class="mt-6 relative z-10">

<div class="px-4 space-y-2">

@for (item of menuItems(); track item.id) {

<button

(click)="setActiveTab(item.id)"

[class]="getButtonClass(item.id)"

>

<i-lucide [name]="item.icon" class="ml-2 h-4 w-4"></i-lucide>

<span>{{item.label}}</span>

</button>
}
</div>
</nav>
</div>
<!-- Main Content -->
<div class="flex-1 overflow-hidden">
<!-- Header -->
<header class="bg-white/80 backdrop-blur-sm shadow-sm border-b border-gray-200 p-4">
<div class="flex items-center justify-between">
<div class="flex items-center space-x-4 space-x-reverse">
<h1 class="text-2xl font-bold bg-gradient-to-r from-purple-600 to-blue-600 bg-clip-text text-transparent">
{{ getPageTitle() }}
</h1>
<span class="px-3 py-1 bg-gradient-to-r from-green-500 to-emerald-500 text-white text-xs rounded-full flex items-center">
<div class="w-2 h-2 bg-white rounded-full ml-1 animate-pulse"></div>
 מחובר
</span>
</div>
<div class="flex items-center space-x-4 space-x-reverse">
<div class="relative">
<i-lucide name="search" class="absolute right-3 top-1/2 transform -translate-y-1/2 text-gray-400 h-4 w-4"></i-lucide>
<input
type="text"
```

```
placeholder="חיפוש..."
class="pr-10 w-64 px-3 py-2 border border-gray-300 rounded-lg focus:border-purple-500 focus:ring-1 focus:ring-purple-500 outline-none transition-colors"
/>
</div>
<button class="p-2 border border-gray-300 rounded-lg hover:bg-gradient-to-r hover:from-purple-50 hover:to-blue-50 transition-all duration-200">
<i-lucide name="bell" class="h-4 w-4"></i-lucide>
</button>
<div class="w-8 h-8 bg-gradient-to-br from-purple-400 to-blue-400 rounded-full flex items-center justify-center text-white font-medium text-sm ring-2 ring-purple-200">
Admin
</div>
</div>
</div>
</header>
<!-- Content Area -->
<main class="p-6">
@switch (activeTab()) {
@case ('dashboard') {
<app-dashboard></app-dashboard>
}
@case ('files') {
<app-files></app-files>
}
@case ('users') {
<app-users></app-users>
}
@case ('categories') {
<app-categories></app-categories>
}
@case ('settings') {
<app-settings></app-settings>
}
}
</main>
</div>
</div>
```

```
import { Routes } from '@angular/router';
import { DashboardComponent } from './components/dashboard/dashboard.component';
export const routes: Routes = [
{ path: ", redirectTo: '/dashboard', pathMatch: 'full' },
{ path: 'dashboard', component: DashboardComponent },
{
path: 'users',
loadComponent: () => import('./components/users/users.component').then(m => m.UsersComponent)
},
{
path: 'files',
loadComponent: () => import('./components/files/files.component').then(m => m.FilesComponent)
},
{
path: 'settings',
loadComponent: () => import('./components/settings/settings.component').then(m => m.SettingsComponent)
}
];
```