// src/app/models/user.model.ts

export interface UserTaxes {

  federal\_tax\_rate: number;

  state\_tax\_rate: number;

  local\_tax\_rate: number;

}

export interface Symbol {

  symbol: string;

  award\_price: number;

  granted: number;

  grant\_date: string;

  exercisable?: number;

  discounted\_price?: number;

  vested?: number;

  unvested?: number;

  next\_vesting\_date?: string;

  vesting\_schedule?: VestingScheduleItem[];

}

export interface VestingScheduleItem {

  date: string;

  amount: number;

  status: 'pending' | 'complete';

}

export interface Award {

  type: 'rsu' | 'esop';

  symbols: Symbol[];

}

export interface User {

  emp\_id: string;

  name: string;

  email: string;

  pwd: string;

  taxes: UserTaxes;

  awards: Award[];

}

// src/app/services/auth.service.ts

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http';

import { BehaviorSubject, Observable, of, throwError } from 'rxjs';

import { tap, catchError } from 'rxjs/operators';

import { User } from '../models/user.model';

@Injectable({

  providedIn: 'root'

})

export class AuthService {

  private currentUserSubject = new BehaviorSubject<User | null>(null);

  public currentUser$ = this.currentUserSubject.asObservable();

  constructor(private http: HttpClient) {

    const userData = localStorage.getItem('currentUser');

    if (userData) {

      this.currentUserSubject.next(JSON.parse(userData));

    }

  }

  // Simulate HTTP login request

  login(email: string, password: string): Observable<User> {

    // In a real app, this would call an API endpoint

    // For demo purposes, we'll use a mock approach

    return this.http.get<User[]>('assets/db.json')

      .pipe(

        tap(users => {

          // Find matching user

          const user = Array.isArray(users) ?

users.find(u => [u.email](http://u.email) === email && u.pwd === password) :

            (users as any).email === email && (users as any).pwd === password ? users as any : null;

          if (!user) {

            throw new Error('Invalid email or password');

          }

          // Update vesting schedule calculations

          const updatedUser = this.calculateVestingSchedule(user);

          // Store user in localStorage

          localStorage.setItem('currentUser', JSON.stringify(updatedUser));

          this.currentUserSubject.next(updatedUser);

          return updatedUser;

        }),

        catchError(err => {

          return throwError(() => new Error('Invalid email or password'));

        })

      );

  }

  logout(): void {

    localStorage.removeItem('currentUser');

    this.currentUserSubject.next(null);

  }

  get currentUserValue(): User | null {

    return this.currentUserSubject.value;

  }

  private calculateVestingSchedule(user: User): User {

    const updatedUser = { ...user };

    // Calculate quarterly vesting for RSUs and ESOPs

    updatedUser.awards.forEach(award => {

      award.symbols.forEach(symbol => {

        // Default vesting is quarterly (every 3 months) over 4 years

        const grantDate = new Date(symbol.grant\_date);

        const totalVestingPeriods = 16; // 4 years × 4 quarters

        const sharesPerPeriod = Math.floor(symbol.granted / totalVestingPeriods);

        // Calculate vested shares based on current date

        const currentDate = new Date();

        let vestedAmount = 0;

        let nextVestingDate = null;

        const vestingSchedule: VestingScheduleItem[] = [];

        for (let i = 0; i < totalVestingPeriods; i++) {

          const vestingDate = new Date(grantDate);

          vestingDate.setMonth(grantDate.getMonth() + (i \* 3));

          const vestingItem: VestingScheduleItem = {

            date: vestingDate.toISOString().split('T')[0],

            amount: sharesPerPeriod,

            status: currentDate >= vestingDate ? 'complete' : 'pending'

          };

          vestingSchedule.push(vestingItem);

          // Update vested amount if vesting date has passed

          if (currentDate >= vestingDate) {

            vestedAmount += sharesPerPeriod;

          } else if (!nextVestingDate) {

            nextVestingDate = vestingDate;

          }

        }

        // Set calculated values

        symbol.vested = vestedAmount;

        symbol.unvested = symbol.granted - vestedAmount;

        symbol.next\_vesting\_date = nextVestingDate ?

          nextVestingDate.toISOString().split('T')[0] : undefined;

        symbol.vesting\_schedule = vestingSchedule;

      });

    });

    return updatedUser;

  }

}

U.email is for sale at Atom.com!

U.email is a powerful, concise domain name that exudes simplicity and efficiency. The letter U can symbolize you, making it perfect for startups in the email marketing or personal branding industri...

// src/app/services/calculation.service.ts

import { Injectable } from '@angular/core';

import { Symbol, User } from '../models/user.model';

@Injectable({

  providedIn: 'root'

})

export class CalculationService {

  // Mock current stock prices - in a real app, these would come from an API

  private currentStockPrices: Record<string, number> = {

    'AAPL': 100,

    'GOOG': 100,

    'MSFT': 100,

    'AMZN': 100,

    'IBM': 100

  };

  constructor() { }

  getCurrentPrice(symbol: string): number {

    return this.currentStockPrices[symbol] || 100; // Default to 100 if not found

  }

  getVestedAmount(symbol: Symbol): number {

    return symbol.vested || 0;

  }

  getUnvestedAmount(symbol: Symbol): number {

    return symbol.unvested || 0;

  }

  // Calculate current market value based on vested shares and current price

  calculateMarketValue(symbol: Symbol): number {

    const currentPrice = this.getCurrentPrice(symbol.symbol);

    const vestedShares = this.getVestedAmount(symbol);

    return currentPrice \* vestedShares;

  }

  // Calculate total exercisable value

  calculateExercisableValue(symbol: Symbol): number {

    const currentPrice = this.getCurrentPrice(symbol.symbol);

    const exercisable = symbol.exercisable || 0;

    return currentPrice \* exercisable;

  }

  // Calculate time to full vesting in months and days

  calculateTimeToFullVesting(symbol: Symbol): { months: number, days: number } {

    if (!symbol.next\_vesting\_date) {

      return { months: 0, days: 0 };

    }

    const nextVestingDate = new Date(symbol.next\_vesting\_date);

    const currentDate = new Date();

    // Get total days between dates

    const diffTime = Math.abs(nextVestingDate.getTime() - currentDate.getTime());

    const diffDays = Math.ceil(diffTime / (1000 \* 60 \* 60 \* 24));

    const months = Math.floor(diffDays / 30);

    const days = diffDays % 30;

    return { months, days };

  }

  // Format the next vesting date (e.g., Aug 30, 2025)

  formatNextVestingDate(dateString?: string): string {

    if (!dateString) return 'N/A';

    const date = new Date(dateString);

    const month = date.toLocaleString('default', { month: 'short' });

    const day = date.getDate();

    const year = date.getFullYear();

    return `${month} ${day}, ${year}`;

  }

  // Calculate vesting percentage

  calculateVestingPercentage(symbol: Symbol): number {

    if (!symbol.granted || symbol.granted === 0) return 0;

    return Math.round((this.getVestedAmount(symbol) / symbol.granted) \* 100);

  }

  // Calculate tax for a given award

  calculateTax(value: number, user: User): number {

    const totalTaxRate = (user.taxes.federal\_tax\_rate +

                          user.taxes.state\_tax\_rate +

                          user.taxes.local\_tax\_rate) / 100;

    return value \* totalTaxRate;

  }

}

// src/app/guards/auth.guard.ts

import { inject } from '@angular/core';

import { Router } from '@angular/router';

import { AuthService } from '../services/auth.service';

export function authGuard() {

  const authService = inject(AuthService);

  const router = inject(Router);

  if (authService.currentUserValue) {

    return true;

  }

  // Not logged in, redirect to login page

  router.navigate(['/login']);

  return false;

}

// src/app/app.routes.ts

import { Routes } from '@angular/router';

import { LoginComponent } from './components/login/login.component';

import { DashboardComponent } from './components/dashboard/dashboard.component';

import { RsuPageComponent } from './components/rsu-page/rsu-page.component';

import { EsopPageComponent } from './components/esop-page/esop-page.component';

import { VestPageComponent } from './components/vest-page/vest-page.component';

import { AboutPageComponent } from './components/about-page/about-page.component';

import { FeaturesPageComponent } from './components/features-page/features-page.component';

import { NewsPageComponent } from './components/news-page/news-page.component';

import { FaqPageComponent } from './components/faq-page/faq-page.component';

import { GlossaryPageComponent } from './components/glossary-page/glossary-page.component';

import { TaxInfoPageComponent } from './components/tax-info-page/tax-info-page.component';

import { LegalPageComponent } from './components/legal-page/legal-page.component';

import { FeedbackPageComponent } from './components/feedback-page/feedback-page.component';

import { SupportPageComponent } from './components/support-page/support-page.component';

import { authGuard } from './guards/auth.guard';

export const routes: Routes = [

  { path: '', redirectTo: '/login', pathMatch: 'full' },

  { path: 'login', component: LoginComponent },

  {

    path: 'dashboard',

    component: DashboardComponent,

    canActivate: [authGuard],

    children: [

      { path: '', redirectTo: 'rsu', pathMatch: 'full' },

      { path: 'rsu', component: RsuPageComponent },

      { path: 'esop', component: EsopPageComponent },

      { path: 'vest', component: VestPageComponent },

      { path: 'about', component: AboutPageComponent },

      { path: 'features', component: FeaturesPageComponent },

      { path: 'news', component: NewsPageComponent },

      { path: 'faq', component: FaqPageComponent },

      { path: 'glossary', component: GlossaryPageComponent },

      { path: 'tax-info', component: TaxInfoPageComponent },

      { path: 'legal', component: LegalPageComponent },

      { path: 'feedback', component: FeedbackPageComponent },

      { path: 'support', component: SupportPageComponent }

    ]

  },

  { path: '\*\*', redirectTo: '/login' }

];

// src/app/app.config.ts

import { ApplicationConfig } from '@angular/core';

import { provideRouter } from '@angular/router';

import { provideHttpClient } from '@angular/common/http';

import { provideAnimations } from '@angular/platform-browser/animations';

import { routes } from './app.routes';

export const appConfig: ApplicationConfig = {

  providers: [

    provideRouter(routes),

    provideHttpClient(),

    provideAnimations()

  ]

};

// src/app/components/login/login.component.ts

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

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

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

import { Router } from '@angular/router';

import { AuthService } from '../../services/auth.service';

@Component({

  selector: 'app-login',

  standalone: true,

  imports: [CommonModule, FormsModule],

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

  styleUrls: ['./login.component.scss']

})

export class LoginComponent {

  email: string = '';

  password: string = '';

  errorMessage: string = '';

  isLoading: boolean = false;

  showSuccessMessage: boolean = false;

  constructor(

    private authService: AuthService,

    private router: Router

  ) {}

  login(): void {

    this.isLoading = true;

    this.errorMessage = '';

this.authService.login([this.email](http://this.email), this.password)

      .subscribe({

        next: () => {

          this.showSuccessMessage = true;

          setTimeout(() => {

            this.router.navigate(['/dashboard']);

          }, 3000);

        },

        error: (error) => {

          this.errorMessage = error.message;

          this.isLoading = false;

        }

      });

  }

}

<!-- src/app/components/login/login.component.html -->

<div class="login-container">

  <div class="login-card">

    <div class="login-header">

      <div class="logo">

        <img src="assets/images/equihub-logo.png" alt="EquiHub Logo">

        <h1>EquiHub</h1>

      </div>

      <p>Your Equity Management Platform</p>

    </div>

    <form (ngSubmit)="login()" class="login-form">

      <div class="form-group">

        <label for="email">Email</label>

        <input

          type="email"

          id="email"

          name="email"

          [(ngModel)]="email"

          required

          placeholder="Enter your email"

        >

      </div>

      <div class="form-group">

        <label for="password">Password</label>

        <input

          type="password"

          id="password"

          name="password"

          [(ngModel)]="password"

          required

          placeholder="Enter your password"

        >

      </div>

      <p class="error-message" \*ngIf="errorMessage">{{ errorMessage }}</p>

      <button

        type="submit"

        class="login-button"

        [disabled]="isLoading || !email || !password"

      >

        {{ isLoading ? 'Logging in...' : 'Login' }}

      </button>

    </form>

  </div>

  <div class="success-message" \*ngIf="showSuccessMessage">

    <div class="success-content">

      <span class="check-icon">✓</span>

      <p>Login Successful!</p>

    </div>

  </div>

</div>

/\* src/app/components/login/login.component.scss \*/

.login-container {

  display: flex;

  justify-content: center;

  align-items: center;

  height: 100vh;

  background-color: #f5f5f5;

  font-family: Arial, sans-serif;

  position: relative;

}

.login-card {

  background: white;

  border-radius: 8px;

  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);

  width: 400px;

  padding: 2rem;

}

.login-header {

  text-align: center;

  margin-bottom: 2rem;

  .logo {

    display: flex;

    align-items: center;

    justify-content: center;

    margin-bottom: 1rem;

    img {

      height: 40px;

      margin-right: 12px;

    }

    h1 {

      color: #004d99;

      margin: 0;

      font-weight: 600;

    }

  }

  p {

    color: #666;

    margin: 0;

  }

}

.login-form {

  .form-group {

    margin-bottom: 1.5rem;

    label {

      display: block;

      margin-bottom: 0.5rem;

      font-weight: 500;

      color: #333;

    }

    input {

      width: 100%;

      padding: 0.75rem;

      border: 1px solid #ddd;

      border-radius: 4px;

      font-size: 16px;

      &:focus {

        outline: none;

        border-color: #0066cc;

        box-shadow: 0 0 0 2px rgba(0, 102, 204, 0.2);

      }

    }

  }

  .error-message {

    color: #d32f2f;

    margin-bottom: 1rem;

    font-size: 14px;

  }

  .login-button {

    width: 100%;

    padding: 0.75rem;

    background-color: #0066cc;

    color: white;

    border: none;

    border-radius: 4px;

    font-size: 16px;

    font-weight: 500;

    cursor: pointer;

    transition: background-color 0.2s;

    &:hover {

      background-color: #004d99;

    }

    &:disabled {

      background-color: #99c2ff;

      cursor: not-allowed;

    }

  }

}

.success-message {

  position: fixed;

  top: 0;

  left: 0;

  right: 0;

  bottom: 0;

  background-color: rgba(0, 0, 0, 0.5);

  display: flex;

  justify-content: center;

  align-items: center;

  z-index: 1000;

  .success-content {

    background-color: white;

    padding: 2rem;

    border-radius: 8px;

    display: flex;

    flex-direction: column;

    align-items: center;

    animation: fadeIn 0.3s ease-out;

    .check-icon {

      background-color: #4caf50;

      color: white;

      width: 50px;

      height: 50px;

      border-radius: 50%;

      display: flex;

      justify-content: center;

      align-items: center;

      font-size: 24px;

      margin-bottom: 1rem;

    }

    p {

      font-size: 18px;

      font-weight: 500;

      margin: 0;

      color: #333;

    }

  }

}

@keyframes fadeIn {

  from {

    opacity: 0;

    transform: translateY(-20px);

  }

  to {

    opacity: 1;

    transform: translateY(0);

  }

}

// src/app/components/dashboard/dashboard.component.ts

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

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

import { RouterModule, Router, NavigationEnd } from '@angular/router';

import { filter } from 'rxjs/operators';

import { AuthService } from '../../services/auth.service';

import { User } from '../../models/user.model';

@Component({

  selector: 'app-dashboard',

  standalone: true,

  imports: [CommonModule, RouterModule],

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

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

})

export class DashboardComponent implements OnInit {

  currentUser: User | null = null;

  activeTab: string = 'rsu';

  constructor(

    private authService: AuthService,

    private router: Router

  ) {}

  ngOnInit(): void {

    this.currentUser = this.authService.currentUserValue;

    if (!this.currentUser) {

      this.router.navigate(['/login']);

      return;

    }

    // Update active tab based on current route

[this.router.events](http://this.router.events).pipe(

      filter(event => event instanceof NavigationEnd)

    ).subscribe((event: any) => {

      const url = event.urlAfterRedirects;

      if (url.includes('/dashboard/rsu')) {

        this.activeTab = 'rsu';

      } else if (url.includes('/dashboard/esop')) {

        this.activeTab = 'esop';

      } else if (url.includes('/dashboard/vest')) {

        this.activeTab = 'vest';

      }

    });

  }

  logout(): void {

    this.authService.logout();

    this.router.navigate(['/login']);

  }

}

<!-- src/app/components/dashboard/dashboard.component.html -->

<div class="dashboard-container">

  <div class="sidebar">

    <div class="logo-container">

      <img src="assets/images/equihub-logo.png" alt="EquiHub" class="logo" />

      <span class="logo-text">EquiHub</span>

    </div>

    <div class="nav-links">

      <a

        [routerLink]="['/dashboard/rsu']"

        routerLinkActive="active"

        [ngClass]="{'active-rsu': activeTab === 'rsu'}"

        (click)="activeTab = 'rsu'"

        class="nav-link"

      >

        RSU

      </a>

      <a

        [routerLink]="['/dashboard/esop']"

        routerLinkActive="active"

        [ngClass]="{'active-esop': activeTab === 'esop'}"

        (click)="activeTab = 'esop'"

        class="nav-link"

      >

        ESOP

      </a>

      <a

        [routerLink]="['/dashboard/vest']"

        routerLinkActive="active"

        [ngClass]="{'active-vest': activeTab === 'vest'}"

        (click)="activeTab = 'vest'"

        class="nav-link"

      >

        Vest

      </a>

    </div>

    <button class="logout-btn" (click)="logout()">Logout</button>

  </div>

  <div class="content-container">

    <router-outlet></router-outlet>

    <div class="footer">

      <div class="footer-section">

        <h4>EquiHub</h4>

        <a [routerLink]="['/dashboard/about']">About</a>

        <a [routerLink]="['/dashboard/features']">Key Features</a>

        <a [routerLink]="['/dashboard/news']">News & Updates</a>

        <a>Equity Overview</a>

      </div>

      <div class="footer-section">

        <h4>Help & Reference</h4>

        <a [routerLink]="['/dashboard/faq']">FAQs</a>

        <a [routerLink]="['/dashboard/glossary']">Glossary of Terms</a>

      </div>

      <div class="footer-section">

        <h4>Important Information</h4>

        <a [routerLink]="['/dashboard/tax-info']">Tax Information</a>

        <a [routerLink]="['/dashboard/legal']">Legal Notices</a>

      </div>

      <div class="footer-section">

        <h4>Connect & Contribute</h4>

        <a [routerLink]="['/dashboard/feedback']">Suggestions & Feedback</a>

        <a>Accessibility Feedback</a>

        <a [routerLink]="['/dashboard/support']">Contact Support</a>

      </div>

      <div class="social-links">

        <a href="#" class="social-icon">

          <i class="fa fa-facebook"></i>

        </a>

        <a href="#" class="social-icon">

          <i class="fa fa-instagram"></i>

        </a>

        <a href="#" class="social-icon">

          <i class="fa fa-envelope"></i>

        </a>

        <a href="#" class="social-icon">

          <i class="fa fa-twitter"></i>

        </a>

        <a href="#" class="social-icon">

          <i class="fa fa-linkedin"></i>

        </a>

      </div>

    </div>

    <div class="copyright">

      Copyright© 2025 EquiHub. All Rights Reserved.

    </div>

  </div>

</div>

/\* src/app/components/dashboard/dashboard.component.scss \*/

.dashboard-container {

  display: flex;

  height: 100vh;

  overflow: hidden;

  font-family: Arial, sans-serif;

}

.sidebar {

  width: 120px;

  background-color: #f5f5f5;

  display: flex;

  flex-direction: column;

  align-items: center;

  padding: 20px 0;

  border-right: 1px solid #ddd;

}

.logo-container {

  display: flex;

  flex-direction: column;

  align-items: center;

  margin-bottom: 30px;

  .logo {

    width: 60px;

    height: 60px;

    margin-bottom: 10px;

  }

  .logo-text {

    font-size: 18px;

    font-weight: 600;

    color: #333;

  }

}

.nav-links {

  display: flex;

  flex-direction: column;

  width: 100%;

  .nav-link {

    padding: 12px 0;

    text-align: center;

    text-decoration: none;

    color: #666;

    font-weight: 500;

    margin-bottom: 10px;

    transition: all 0.3s ease;

    border-radius: 4px;

    &:hover {

      background-color: #e6e6e6;

    }

    &.active {

      font-weight: 600;

    }

    &.active-rsu {

      background-color: #0066cc;

      color: white;

    }

    &.active-esop {

      background-color: #009933;

      color: white;

    }

    &.active-vest {

      background-color: #0099cc;

      color: white;

    }

  }

}

.logout-btn {

  margin-top: auto;

  padding: 10px 20px;

/\* src/app/components/dashboard/dashboard.component.scss (continued) \*/

.logout-btn {

  margin-top: auto;

  padding: 10px 20px;

  background-color: #f0f0f0;

  border: 1px solid #ddd;

  border-radius: 4px;

  color: #666;

  font-weight: 500;

  cursor: pointer;

  transition: all 0.3s ease;

  &:hover {

    background-color: #e0e0e0;

  }

}

.content-container {

  flex: 1;

  display: flex;

  flex-direction: column;

  overflow-y: auto;

}

.footer {

  background-color: #00305b;

  color: white;

  padding: 30px 50px;

  display: flex;

  flex-wrap: wrap;

  justify-content: space-between;

  margin-top: auto;

  .footer-section {

    display: flex;

    flex-direction: column;

    margin-right: 50px;

    margin-bottom: 20px;

    h4 {

      margin-top: 0;

      margin-bottom: 15px;

      font-size: 16px;

      font-weight: 600;

    }

    a {

      color: #ccc;

      margin-bottom: 8px;

      text-decoration: none;

      font-size: 14px;

      cursor: pointer;

      &:hover {

        color: white;

        text-decoration: underline;

      }

    }

  }

  .social-links {

    display: flex;

    margin-top: 20px;

    .social-icon {

      width: 36px;

      height: 36px;

      background-color: rgba(255, 255, 255, 0.1);

      border-radius: 50%;

      display: flex;

      align-items: center;

      justify-content: center;

      margin-right: 10px;

      color: white;

      text-decoration: none;

      transition: background-color 0.3s;

      &:hover {

        background-color: rgba(255, 255, 255, 0.2);

      }

      i {

        font-size: 18px;

      }

    }

  }

}

.copyright {

  background-color: #00244a;

  color: #ccc;

  text-align: center;

  padding: 15px;

  font-size: 14px;

}

// src/app/components/rsu-page/rsu-page.component.ts

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

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

import { AuthService } from '../../services/auth.service';

import { CalculationService } from '../../services/calculation.service';

import { User, Symbol } from '../../models/user.model';

@Component({

  selector: 'app-rsu-page',

  standalone: true,

  imports: [CommonModule],

  templateUrl: './rsu-page.component.html',

  styleUrls: ['./rsu-page.component.scss']

})

export class RsuPageComponent implements OnInit {

  user: User | null = null;

  rsuSymbols: Symbol[] = [];

  totalStocksGranted: number = 0;

  totalVestedStocks: number = 0;

  totalUnvestedStocks: number = 0;

  currentValue: number = 0;

  nextVestingDate: string = '';

  timeToFullVesting: { months: number, days: number } = { months: 0, days: 0 };

  currentStockPrice: number = 100; // Default value

  vestingPercentage: number = 0;

  firstVestingDate: string = '';

  constructor(

    private authService: AuthService,

    private calculationService: CalculationService

  ) {}

  ngOnInit(): void {

    this.user = this.authService.currentUserValue;

    if (!this.user) return;

    // Find RSU awards

    const rsuAward = this.user.awards.find(award => award.type === 'rsu');

    if (rsuAward) {

      this.rsuSymbols = rsuAward.symbols;

      // Calculate totals

      this.calculateTotals();

    }

  }

  calculateTotals(): void {

    this.totalStocksGranted = this.rsuSymbols.reduce((total, symbol) => total + symbol.granted, 0);

    this.totalVestedStocks = this.rsuSymbols.reduce((total, symbol) => total + (symbol.vested || 0), 0);

    this.totalUnvestedStocks = this.totalStocksGranted - this.totalVestedStocks;

    // Calculate current value

    this.currentValue = this.rsuSymbols.reduce((total, symbol) => {

      return total + this.calculationService.calculateMarketValue(symbol);

    }, 0);

    // Find next vesting date

    const nextDates = this.rsuSymbols

      .map(symbol => symbol.next\_vesting\_date)

      .filter(date => !!date)

      .sort();

    if (nextDates.length > 0) {

      this.nextVestingDate = this.calculationService.formatNextVestingDate(nextDates[0]);

      // Find first symbol with this vesting date

      const symbolWithNextVesting = this.rsuSymbols.find(

        symbol => symbol.next\_vesting\_date === nextDates[0]

      );

      if (symbolWithNextVesting) {

        this.timeToFullVesting = this.calculationService.calculateTimeToFullVesting(symbolWithNextVesting);

      }

    }

    // Calculate vesting percentage

    if (this.totalStocksGranted > 0) {

      this.vestingPercentage = Math.round((this.totalVestedStocks / this.totalStocksGranted) \* 100);

    }

    // Find first vesting date

    if (this.rsuSymbols.length > 0 && this.rsuSymbols[0].vesting\_schedule && this.rsuSymbols[0].vesting\_schedule.length > 0) {

      this.firstVestingDate = this.calculationService.formatNextVestingDate(this.rsuSymbols[0].vesting\_schedule[0].date);

    }

  }

}

<!-- src/app/components/rsu-page/rsu-page.component.html -->

<div class="rsu-container">

  <div class="header">

    <div class="title">

      <h1>RSU</h1>

    </div>

    <div class="stock-price">Current Stock Price: <span class="price-value">${{ currentStockPrice }}</span></div>

    <div class="user-info">

      <span>{{ user?.name }}</span>

      <img src="assets/images/user-avatar.png" alt="User Avatar" class="avatar">

    </div>

  </div>

  <div class="data-grid">

    <div class="data-card">

      <h3>Total Stocks Granted</h3>

      <div class="card-value">{{ totalStocksGranted }}</div>

    </div>

    <div class="data-card">

      <h3>Total Vested Stocks</h3>

      <div class="card-value">{{ totalVestedStocks }}</div>

    </div>

    <div class="data-card">

      <h3>Remaining Unvested Stocks</h3>

      <div class="card-value">{{ totalUnvestedStocks }}</div>

    </div>

    <div class="data-card">

      <h3>My Current Value</h3>

      <div class="card-value">${{ currentValue }}</div>

    </div>

    <div class="data-card">

      <h3>Next Vesting Date</h3>

      <div class="card-value">{{ nextVestingDate }}</div>

    </div>

  </div>

  <div class="widget-grid">

    <div class="widget">

      <h3>Grant Agreements</h3>

      <button class="download-btn">Download</button>

    </div>

    <div class="widget">

      <h3>Vesting Progress</h3>

      <div class="progress-circle">

        <svg viewBox="0 0 36 36" class="circular-chart">

          <path class="circle-bg"

            d="M18 2.0845

              a 15.9155 15.9155 0 0 1 0 31.831

              a 15.9155 15.9155 0 0 1 0 -31.831"

          />

          <path class="circle"

            [attr.stroke-dasharray]="vestingPercentage + ', 100'"

            d="M18 2.0845

              a 15.9155 15.9155 0 0 1 0 31.831

              a 15.9155 15.9155 0 0 1 0 -31.831"

          />

          <text x="18" y="20.35" class="percentage">{{ vestingPercentage }}%</text>

        </svg>

        <div class="vested-label">Vested</div>

      </div>

    </div>

    <div class="widget">

      <h3>Vesting History</h3>

      <div class="vesting-history">

        <div class="history-item">

          <span class="history-label">1st Vest:</span>

          <span class="history-value">{{ firstVestingDate }}</span>

        </div>

      </div>

    </div>

    <div class="widget">

      <h3>Time to Full Vesting</h3>

      <div class="time-to-vesting">

        <span class="time-value">{{ timeToFullVesting.months }}</span>

        <span class="time-label">months,</span>

        <span class="time-value">{{ timeToFullVesting.days }}</span>

        <span class="time-label">days</span>

      </div>

    </div>

    <div class="widget">

      <h3>Tax Calculator</h3>

      <div class="tax-calculator">

        <div class="calculator-fields">

          <div class="field">

            <span>State</span>

          </div>

          <div class="field">

            <span>Rate</span>

          </div>

          <div class="field">

            <span>Total</span>

          </div>

        </div>

        <button class="calculate-btn">Calculate</button>

      </div>

    </div>

  </div>

</div>

/\* src/app/components/rsu-page/rsu-page.component.scss \*/

.rsu-container {

  padding: 20px;

  background-color: #f5f9fd;

  height: 100%;

  overflow-y: auto;

}

.header {

  display: flex;

  justify-content: space-between;

  align-items: center;

  margin-bottom: 20px;

  .title {

    h1 {

      font-size: 28px;

      color: #0066cc;

      margin: 0;

    }

  }

  .stock-price {

    font-size: 16px;

    font-weight: 500;

    .price-value {

      color: #0066cc;

      font-weight: 600;

    }

  }

  .user-info {

    display: flex;

    align-items: center;

    span {

      margin-right: 10px;

      font-weight: 500;

    }

    .avatar {

      width: 36px;

      height: 36px;

      border-radius: 50%;

      object-fit: cover;

    }

  }

}

.data-grid {

  display: grid;

  grid-template-columns: repeat(5, 1fr);

  gap: 20px;

  margin-bottom: 30px;

  .data-card {

    background-color: white;

    border-radius: 8px;

    padding: 15px;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.05);

    text-align: center;

    h3 {

      font-size: 14px;

      color: #666;

      margin-top: 0;

      margin-bottom: 10px;

      font-weight: 500;

    }

    .card-value {

      font-size: 24px;

      font-weight: 600;

      color: #333;

    }

  }

}

.widget-grid {

  display: grid;

  grid-template-columns: repeat(5, 1fr);

  gap: 20px;

  .widget {

    background-color: white;

    border-radius: 8px;

    padding: 20px;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.05);

    h3 {

      font-size: 16px;

      color: #333;

      margin-top: 0;

      margin-bottom: 20px;

      text-align: center;

    }

    .download-btn {

      display: block;

      width: 100%;

      padding: 10px;

      background-color: #00478f;

      color: white;

      border: none;

      border-radius: 4px;

      cursor: pointer;

      transition: background-color 0.3s;

      &:hover {

        background-color: #003366;

      }

    }

  }

}

.progress-circle {

  display: flex;

  flex-direction: column;

  align-items: center;

  .circular-chart {

    display: block;

    margin: 0 auto;

    max-width: 80%;

    max-height: 250px;

  }

  .circle-bg {

    fill: none;

    stroke: #eee;

    stroke-width: 3.8;

  }

  .circle {

    fill: none;

    stroke-width: 2.8;

    stroke-linecap: round;

    stroke: #0066cc;

    animation: progress 1s ease-out forwards;

  }

  .percentage {

    fill: #0066cc;

    font-size: 0.5em;

    text-anchor: middle;

    font-weight: bold;

  }

  .vested-label {

    margin-top: 5px;

    font-size: 14px;

    color: #0066cc;

  }

}

.vesting-history {

  .history-item {

    display: flex;

    justify-content: center;

    margin-bottom: 10px;

    .history-label {

      font-weight: 500;

      margin-right: 5px;

    }

    .history-value {

      color: #0066cc;

    }

  }

}

.time-to-vesting {

  text-align: center;

  .time-value {

    font-size: 24px;

    font-weight: 600;

    color: #0066cc;

    margin-right: 5px;

  }

  .time-label {

    font-size: 16px;

    color: #666;

    margin-right: 10px;

  }

}

.tax-calculator {

  .calculator-fields {

    border: 1px solid #ddd;

    border-radius: 4px;

    margin-bottom: 10px;

    padding: 10px;

    .field {

      display: flex;

      justify-content: space-between;

      margin-bottom: 5px;

      span {

        color: #666;

      }

    }

  }

  .calculate-btn {

    display: block;

    width: 100%;

    padding: 8px;

    background-color: #00478f;

    color: white;

    border: none;

    border-radius: 4px;

    cursor: pointer;

    transition: background-color 0.3s;

    &:hover {

      background-color: #003366;

    }

  }

}

@keyframes progress {

  0% {

    stroke-dasharray: 0 100;

  }

}

// src/app/components/esop-page/esop-page.component.ts

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

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

import { AuthService } from '../../services/auth.service';

import { CalculationService } from '../../services/calculation.service';

import { User, Symbol } from '../../models/user.model';

@Component({

  selector: 'app-esop-page',

  standalone: true,

  imports: [CommonModule],

  templateUrl: './esop-page.component.html',

  styleUrls: ['./esop-page.component.scss']

})

export class EsopPageComponent implements OnInit {

  user: User | null = null;

  esopSymbols: Symbol[] = [];

  totalStocksGranted: number = 0;

  totalVestedStocks: number = 0;

  totalUnvestedStocks: number = 0;

  currentMarketValue: number = 0;

  nextVestingDate: string = '';

  timeToFullVesting: { months: number, days: number } = { months: 0, days: 0 };

  currentStockPrice: number = 100; // Default value

  discountedPrice: number = 80; // Default value

  vestingPercentage: number = 0;

  exercisedStocks: number = 0;

  remainingStocksToExercise: number = 0;

  constructor(

    private authService: AuthService,

    private calculationService: CalculationService

  ) {}

  ngOnInit(): void {

    this.user = this.authService.currentUserValue;

    if (!this.user) return;

    // Find ESOP awards

    const esopAward = this.user.awards.find(award => award.type === 'esop');

    if (esopAward) {

      this.esopSymbols = esopAward.symbols;

      // Calculate totals

      this.calculateTotals();

    }

  }

  calculateTotals(): void {

    this.totalStocksGranted = this.esopSymbols.reduce((total, symbol) => total + symbol.granted, 0);

    this.totalVestedStocks = this.esopSymbols.reduce((total, symbol) => total + (symbol.vested || 0), 0);

    this.totalUnvestedStocks = this.totalStocksGranted - this.totalVestedStocks;

    // Get discounted price from first symbol if available

    if (this.esopSymbols.length > 0 && this.esopSymbols[0].discounted\_price) {

      this.discountedPrice = this.esopSymbols[0].discounted\_price;

    }

    // Calculate current market value

    this.currentMarketValue = this.esopSymbols.reduce((total, symbol) => {

      return total + this.calculationService.calculateMarketValue(symbol);

    }, 0);

    // Find next vesting date

    const nextDates = this.esopSymbols

      .map(symbol => symbol.next\_vesting\_date)

      .filter(date => !!date)

      .sort();

    if (nextDates.length > 0) {

      this.nextVestingDate = this.calculationService.formatNextVestingDate(nextDates[0]);

    }

    // Calculate time to full vesting

    if (this.esopSymbols.length > 0 && this.esopSymbols[0].next\_vesting\_date) {

      this.timeToFullVesting = this.calculationService.calculateTimeToFullVesting(this.esopSymbols[0]);

    }

    // Calculate vesting percentage

    if (this.totalStocksGranted > 0) {

      this.vestingPercentage = Math.round((this.totalVestedStocks / this.totalStocksGranted) \* 100);

    }

    // Calculate exercised stocks and remaining stocks to exercise

    this.exercisedStocks = this.esopSymbols.reduce((total, symbol) => total + (symbol.exercisable || 0), 0);

    this.remainingStocksToExercise = this.totalVestedStocks - this.exercisedStocks;

  }

}

<!-- src/app/components/esop-page/esop-page.component.html -->

<div class="esop-container">

  <div class="header">

    <div class="title">

      <h1>ESOP</h1>

    </div>

    <div class="stock-prices">

      <div class="price-item">Current Stock Price: <span class="price-value">${{ currentStockPrice }}</span></div>

      <div class="price-item">Discounted Price: <span class="price-value discounted">${{ discountedPrice }}</span></div>

    </div>

    <div class="user-info">

      <span>{{ user?.name }}</span>

      <img src="assets/images/user-avatar.png" alt="User Avatar" class="avatar">

    </div>

  </div>

  <div class="data-grid">

    <div class="data-card">

      <h3>Total Stocks Granted</h3>

      <div class="card-value">{{ totalStocksGranted }}</div>

    </div>

    <div class="data-card">

      <h3>Total Vested Stocks</h3>

      <div class="card-value">{{ totalVestedStocks }}</div>

    </div>

    <div class="data-card">

      <h3>Remaining Unvested Stocks</h3>

      <div class="card-value">{{ totalUnvestedStocks }}</div>

    </div>

    <div class="data-card">

      <h3>Current Market Value</h3>

      <div class="card-value">${{ currentMarketValue }}</div>

    </div>

    <div class="data-card">

      <h3>Next Vesting Date</h3>

      <div class="card-value">{{ nextVestingDate }}</div>

    </div>

  </div>

  <div class="widget-grid">

    <div class="data-card">

      <h3>Time to Full Vesting</h3>

      <div class="time-to-vesting">

        <span class="time-value">{{ timeToFullVesting.months }}</span>

        <span class="time-label">months,</span>

        <span class="time-value">{{ timeToFullVesting.days }}</span>

        <span class="time-label">days</span>

      </div>

    </div>

    <div class="data-card">

      <h3>Vesting Progress</h3>

      <div class="progress-circle">

        <svg viewBox="0 0 36 36" class="circular-chart">

          <path class="circle-bg"

            d="M18 2.0845

              a 15.9155 15.9155 0 0 1 0 31.831

              a 15.9155 15.9155 0 0 1 0 -31.831"

          />

          <path class="circle"

            [attr.stroke-dasharray]="vestingPercentage + ', 100'"

            d="M18 2.0845

              a 15.9155 15.9155 0 0 1 0 31.831

              a 15.9155 15.9155 0 0 1 0 -31.831"

          />

          <text x="18" y="20.35" class="percentage">{{ vestingPercentage }}%</text>

        </svg>

        <div class="vested-label">Vested</div>

      </div>

    </div>

    <div class="data-card">

      <h3>Exercised Stocks</h3>

      <div class="card-value">{{ exercisedStocks }}</div>

    </div>

    <div class="data-card">

      <h3>Remaining Stocks to Exercise</h3>

      <div class="card-value">{{ remainingStocksToExercise }}</div>

    </div>

    <div class="data-card">

      <h3>Tax Calculator</h3>

      <div class="tax-calculator">

        <div class="calculator-fields">

          <div class="field">

            <span>State</span>

          </div>

          <div class="field">

            <span>Rate</span>

          </div>

          <div class="field">

            <span>Total</span>

          </div>

        </div>

        <button class="calculate-btn">Calculate</button>

      </div>

    </div>

  </div>

</div>

/\* src/app/components/esop-page/esop-page.component.scss \*/

.esop-container {

  padding: 20px;

  background-color: #f5fdf7;

  height: 100%;

  overflow-y: auto;

}

.header {

  display: flex;

  justify-content: space-between;

  align-items: center;

  margin-bottom: 20px;

  .title {

    h1 {

      font-size: 28px;

      color: #009933;

      margin: 0;

    }

  }

  .stock-prices {

    .price-item {

      font-size: 16px;

      font-weight: 500;

      margin-bottom: 5px;

      .price-value {

        font-weight: 600;

        &.discounted {

          color: #009933;

        }

      }

    }

  }

  .user-info {

    display: flex;

    align-items: center;

    span {

      margin-right: 10px;

      font-weight: 500;

    }

    .avatar {

      width: 36px;

      height: 36px;

      border-radius: 50%;

      object-fit: cover;

    }

  }

}

.data-grid {

  display: grid;

  grid-template-columns: repeat(5, 1fr);

  gap: 20px;

  margin-bottom: 30px;

  .data-card {

    background-color: white;

    border-radius: 8px;

    padding: 15px;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.05);

    text-align: center;

    h3 {

      font-size: 14px;

      color: #666;

      margin-top: 0;

      margin-bottom: 10px;

      font-weight: 500;

    }

    .card-value {

      font-size: 24px;

      font-weight: 600;

      color: #333;

    }

  }

}

.widget-grid {

  display: grid;

  grid-template-columns: repeat(5, 1fr);

  gap: 20px;

  .data-card {

    background-color: white;

    border-radius: 8px;

    padding: 20px;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.05);

    h3 {

      font-size: 16px;

      color: #333;

      margin-top: 0;

      margin-bottom: 20px;

      text-align: center;

    }

  }

}

.time-to-vesting {

  text-align: center;

  .time-value {

    font-size: 24px;

    font-weight: 600;

    color: #009933;

    margin-right: 5px;

  }

  .time-label {

    font-size: 16px;

    color: #666;

    margin-right: 10px;

  }

}

.progress-circle {

  display: flex;

  flex-direction: column;

  align-items: center;

  .circular-chart {

    display: block;

    margin: 0 auto;

    max-width: 80%;

    max-height: 250px;

  }

  .circle-bg {

    fill: none;

    stroke: #eee;

    stroke-width: 3.8;

  }

  .circle {

    fill: none;

    stroke-width: 2.8;

    stroke-linecap: round;

    stroke: #009933;

    animation: progress 1s ease-out forwards;

  }

  .percentage {

    fill: #009933;

    font-size: 0.5em;

    text-anchor: middle;

    font-weight: bold;

  }

  .vested-label {

    margin-top: 5px;

    font-size: 14px;

    color: #009933;

  }

}

.tax-calculator {

  .calculator-fields {

    border: 1px solid #ddd;

    border-radius: 4px;

    margin-bottom: 10px;

    padding: 10px;

    .field {

      display: flex;

      justify-content: space-between;

      margin-bottom: 5px;

      span {

        color: #666;

      }

    }

  }

  .calculate-btn {

    display: block;

    width: 100%;

    padding: 8px;

    background-color: #009933;

    color: white;