Dashboard.component.html

<!-- main wrapper for the entire dashboard-->

<div class="dashboard-container">

<!-- Header -->

<div class="header">

<div class="title">

<h1>RSU</h1>

</div>

<!-- User Info -->

<div class="stock-info">

<!-- Stcok price -->

<span>Current Stock Price : </span>

<!--\*ngIf ensures it renders only if equityData is loaded.-->

<span class="stock-price" \*ngIf="equityData">${{equityData.stockPrice}}</span>

</div>

<!--Displays a user avatar placeholder.-->

<div class="user-info">

<span \*ngIf="equityData">{{equityData.username}}</span>

<div class="user-avatar">

<div class="avatar-placeholder"></div>

</div>

</div>

</div>

<!-- colored background (blue) for the main section.-->

<div class="blue-container">

<div class="dashboard-content" \*ngIf="!loading && !error">

<div class="row g-4">

<div class="col-md-6">

<!--Each app-stock-card or app-vesting-progress is a reusable

Angular component passed props ([value], [title], etc.).-->

<app-stock-card

title="Total Stocks Granted"

[value]="equityData?.totalStocksGranted || 0"

colorClass="dark-blue"

class="top-left-card">

</app-stock-card>

</div>

<div class="col-md-6">

<!--Shows a progress bar and next vesting date-->

<app-vesting-progress

[percentage]="equityData?.vestingPercentage || 0"

[nextDate]="equityData?.nextVestingDate || ''">

</app-vesting-progress>

</div>

<div class="col-md-6">

<!--Shows the remaining stocks that haven’t vested yet-->

<app-stock-card

title="Remaining Unvested Stocks"

[value]="equityData?.remainingUnvestedStocks || 0"

colorClass="teal"

class="bottom-left-card">

</app-stock-card>

</div>

<div class="col-md-6">

<app-stock-card

title="My Current Value"

[value]="equityData?.currentValue || 0"

[isCurrency]="true"

colorClass="dark-blue"

class="bottom-right-card">

</app-stock-card>

</div>

</div>

</div>

<!--Shows a Bootstrap spinner while data is loading-->

<div class="loading-container" \*ngIf="loading">

<div class="spinner-border text-primary" role="status">

<span class="visually-hidden">Loading...</span>

</div>

</div>

<!--If there's an error fetching data, this message is shown-->

<div class="error-container" \*ngIf="error">

<div class="alert alert-danger">

Error loading equity data. Please try again later.

</div>

</div>

</div>

</div>

<!-- Footer Label -->

<div class="footer-label">

© 2025 EquiHub. All Rights Reserved.

</div>

Dashboard.component.scss

.dashboard-container {

background-color: white;//bg for entire dashboard

min-height: 100vh;//container takes full screen height

padding: 20px;//: Adds space inside the container

display: flex;

flex-direction: column;//Main layout in vertical flow

flex-wrap: wrap;// Useful if you expect child elements to wrap

gap: 20px; //space between child elements

justify-content: space-between;//distributes space between child elements

@media (max-width: 768px) {

flex-direction: column; //// On small screens, maintain vertical layout

align-items: center;// Horizontally center child elements

padding: 10px !important;// Reduces padding on smaller screens and overrides other styles

}

}

//Used to arrange title and user info on the same row, spacing them out horizontally.

.title-user-container {

display: flex;

justify-content: space-between;// Space out title and user info on the same row

align-items: center;// Align both title and user info vertically center

width: 100%; // Make it span the full width of the parent container

}

// Header section containing RSU title and user info

.header {

display: flex;

justify-content: space-between;// Title and user info sit on opposite ends

align-items: center;// Vertically center content

padding-bottom: 20px;// Add space below the header for visual separation

//Styled RSU heading

.title h1 {

color: #0b476b;

font-weight: bold; // Makes the heading bold

margin: 0;// Removes default margin

font-size: 2.5rem; // Makes the title large

}

//Shows stock information like stock price and stock name

.stock-info {

font-size: 1.9rem;

text-align: center; // Centers the text inside

//This is the stock price value

.stock-price {

color: #00a651;

text-align: center; // Centers the stock price

font-weight: bold;// Highlights it with bold font

font-size: 1.7rem; // Slightly smaller than main stock info

}

}

// User profile section including avatar and name

.user-info {

display: flex;// Layout items side by side

align-items: center;

gap: 10px;

span {

font-weight: 500;// Semi-bold username

}

// Circle container for user's avatar

.user-avatar {

width: 40px;

height: 40px;

background-color: #ccc; // Gray fallback color

border-radius: 50%;

overflow: hidden; // Makes it a perfect circle

// Placeholder image inside the avatar

.avatar-placeholder {

width: 100%; // Takes full width of the parent

height: 100%;

background-color: #888;

}

}

}

}

//Container for the main content of the dashboard

// Section with light blue background for main content area

.blue-container {

background-color: #c3dcf7;

border-radius: 16px;//rounded corners

padding: 20px;// Space inside the container

margin-bottom: 20px;// Space below the container to separate it from other elements

flex-grow: 1;//// Allows the container to grow and fill available space in Flexbox

}

// This is the main content area of the dashboard

.dashboard-content {

height: 100%;

}

// This is the container for the cards in the dashboard

.loading-container, .error-container {

display: flex;

justify-content: center;

align-items: center;

min-height: 200px;//// Minimum height so it doesn't collapse while loading

}

/\* Top-left card \*/

:host ::ng-deep .top-left-card .stock-card {

height: 360px;// Fixed height for consistency

border: 2px solid black;

border-radius: 8px; // Slightly rounded corners

background-color: white;

}

:host ::ng-deep .bottom-left-card .stock-card {

height: 260px; /\* Increase this value as needed \*/

border: 2px solid black;

position: absolute;

width: 97%;

bottom: 0; /\* This anchors the card to the bottom \*/

}

/\* Make the container have relative positioning \*/

.col-md-6 {

position: relative;

min-height: 200px; /\* Should match or exceed card height \*/

}

/\* Top-right card (vesting progress) - maintaining original styling \*/

:host ::ng-deep .vesting-card {

height: 200px;

border: 2px solid black;

}

/\* Bottom-right card \*/

:host ::ng-deep .bottom-right-card .stock-card {

height: 200px;

border: 2px solid black;

border-radius: 8px;

background-color: white;

}

/\* Common styling for all cards \*/

:host ::ng-deep .stock-card,

:host ::ng-deep .vesting-card {

border: 2px solid black;

padding: 15px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

margin-bottom: 15px;

}

.footer-label {

width: 100%;

background-color: black;

color: white;

text-align: center;

padding: 10px 0;

font-weight: 500;

font-size: 1rem;

position: fixed;

bottom: 0;

left: 0;

z-index: 999; // Stays above other content

}

/\* Mobile-specific adjustments \*/

@media (max-width: 768px) {

.dashboard-container {

padding: 10px;

}

.blue-container {

padding: 15px;

}

.header {

display: flex;

flex-direction: column;

align-items: flex-start;

padding-top: 20px;

}

.title h1 {

font-size: 1.5rem;

}

.top-row {

display: flex;

justify-content: space-between;

align-items: center;

width: 100%;

}

.stock-info {

margin-top: 50px;/\* Increase this value to move stock info further down \*/

font-size: 0.9rem;

width: 100%;

order:2;

display: flex; /\* Use flexbox inside stock-info \*/

justify-content: center; /\* Centers the stock info horizontally \*/

align-items: center;

.stock-price {

font-size: 0.8rem;

order: 2;

}

}

/\* Keep RSU title and user info on same line \*/

.title, .user-info {

display: flex !important;

align-items: center;

justify-content: space-between;

margin-bottom: 1px;

}

.row.g-4 {

display: flex;

flex-wrap: wrap;

margin: 0 -8px;

}

.col-md-6 {

flex: 0 0 50% !important;

width: 50% !important;

padding: 0 8px !important;

min-height: 180px;

}

/\* Fix card heights \*/

:host ::ng-deep .top-left-card .stock-card,

:host ::ng-deep .bottom-left-card .stock-card,

:host ::ng-deep .bottom-right-card .stock-card,

:host ::ng-deep .vesting-card {

height: auto !important;

min-height: 180px;

width: 100% !important;

position: relative !important;

bottom: auto;

}

/\* Adjust card text sizes \*/

:host ::ng-deep .stock-card .card-title,

:host ::ng-deep .vesting-card .card-title {

font-size: 1.5rem !important;

margin-bottom: 10px !important;

}

:host ::ng-deep .stock-card .card-value {

font-size: 2rem !important;

}

/\* Fix vesting progress chart size \*/

:host ::ng-deep .vesting-card .chart-container {

width: 120px !important;

height: 120px !important;

}

:host ::ng-deep .vesting-card .percentage-indicator .percentage {

font-size: 1.5rem !important;

}

:host ::ng-deep .vesting-card .next-vesting-date {

padding: 8px 20px !important;

}

:host ::ng-deep .vesting-card .next-vesting-date .date-label,

:host ::ng-deep .vesting-card .next-vesting-date .date-value {

font-size: 1rem !important;

}

.dashboard-container {

padding-bottom: 60px; // add extra bottom space

}

}

Dashboard.component.ts

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

import { EquityService, EquityData } from '../../services/equity.service';

@Component({

selector: 'app-dashboard',

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

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

})

export class DashboardComponent implements OnInit {

equityData: EquityData | null = null; //Holds RSU details fetched from a backend API

loading = true;//True while data is being fetched.

error = false;//True if fetching data fails

constructor(private equityService: EquityService) {}

ngOnInit(): void { // Angular lifecycle hook — triggers loadData() when the component loads.

this.loadData();

}

loadData(): void { //Uses EquityService to get the data, and updates UI state based on success or failure.

this.equityService.getEquityData().subscribe({

next: (data) => {

this.equityData = data;

this.loading = false;

},

error: (err) => {

console.error('Error loading equity data:', err);

this.error = true;

this.loading = false;

}

});

}

}

Esop-dashboard.component.html

<!-- esop-dashboard.component.html -->

<!-- main wrapper for the entire dashboard-->

<div class="dashboard-container">

<!-- Header -->

<div class="header">

<div class="title">

<h1>ESOP</h1>

</div>

<!-- Stock Info -->

<div class="stock-info">

<!-- Stock price -->

<span>Current Stock Price : </span>

<!--\*ngIf ensures it renders only if equityData is loaded.-->

<span class="stock-price" \*ngIf="equityData">${{equityData.stockPrice}}</span>

</div>

<!--Displays a user avatar placeholder.-->

<div class="user-info">

<span \*ngIf="equityData">{{equityData.username}}</span>

<div class="user-avatar">

<div class="avatar-placeholder"></div>

</div>

</div>

</div>

<!-- colored background (light green) for the main section.-->

<div class="green-container">

<div class="dashboard-content" \*ngIf="!loading && !error">

<div class="row g-4">

<div class="col-md-6">

<!--Each app-stock-card or app-vesting-progress is a reusable

Angular component passed props ([value], [title], etc.).-->

<app-stock-card

title="Total Stocks granted"

[value]="equityData?.totalStocksGranted || 0"

colorClass="dark-blue"

class="top-left-card">

</app-stock-card>

</div>

<div class="col-md-6">

<!--Shows a progress bar and next vesting date-->

<app-esop-vesting-progress

[percentage]="equityData?.vestingPercentage || 0"

[nextDate]="'Amount Needed to Exercise'"

[amountValue]="500">

</app-esop-vesting-progress>

</div>

<div class="col-md-6">

<!--Shows the remaining stocks that haven't vested yet-->

<app-stock-card

title="Remaining Unvested"

[value]="equityData?.remainingUnvestedStocks || 0"

colorClass="teal"

class="bottom-left-card">

</app-stock-card>

</div>

<div class="col-md-6">

<app-stock-card

title="Potential Profit If Exercised Today"

[value]="2000"

[isCurrency]="true"

colorClass="teal"

class="bottom-right-card">

</app-stock-card>

</div>

</div>

</div>

<!--Shows a Bootstrap spinner while data is loading-->

<div class="loading-container" \*ngIf="loading">

<div class="spinner-border text-primary" role="status">

<span class="visually-hidden">Loading...</span>

</div>

</div>

<!--If there's an error fetching data, this message is shown-->

<div class="error-container" \*ngIf="error">

<div class="alert alert-danger">

Error loading equity data. Please try again later.

</div>

</div>

</div>

</div>

<div class="footer-label">

© 2025 EquiHub. All Rights Reserved.

</div>

Esop-dashboard.component.scss

// esop-dashboard.component.scss

.dashboard-container {

background-color: white;//bg for entire dashboard

min-height: 100vh;//container takes full screen height

padding: 20px;//: Adds space inside the container

display: flex;

flex-direction: column;//Main layout in vertical flow

flex-wrap: wrap;// Useful if you expect child elements to wrap

gap: 20px; //space between child elements

justify-content: space-between;//distributes space between child elements

@media (max-width: 768px) {

flex-direction: column; //// On small screens, maintain vertical layout

align-items: center;// Horizontally center child elements

padding: 10px !important;// Reduces padding on smaller screens and overrides other styles

}

}

//Used to arrange title and user info on the same row, spacing them out horizontally.

.title-user-container {

display: flex;

justify-content: space-between;// Space out title and user info on the same row

align-items: center;// Align both title and user info vertically center

width: 100%; // Make it span the full width of the parent container

}

// Header section containing ESOP title and user info

.header {

display: flex;

justify-content: space-between;// Title and user info sit on opposite ends

align-items: center;// Vertically center content

padding-bottom: 20px;// Add space below the header for visual separation

//Styled ESOP heading

.title h1 {

color: #135e13; // Dark green color for ESOP

font-weight: bold; // Makes the heading bold

margin: 0;// Removes default margin

font-size: 2.5rem; // Makes the title large

}

//Shows stock information like stock price and stock name

.stock-info {

display: flex;

flex-direction: column;

align-items: center;// Centers the text inside

//This is the stock price value

.stock-price {

color: #00a651;

text-align: center; // Centers the stock price

font-weight: bold;// Highlights it with bold font

font-size: 1.7rem; // Slightly smaller than main stock info

}

}

// User profile section including avatar and name

.user-info {

display: flex;// Layout items side by side

align-items: center;

gap: 10px;

span {

font-weight: 500;// Semi-bold username

}

// Circle container for user's avatar

.user-avatar {

width: 40px;

height: 40px;

background-color: #ccc; // Gray fallback color

border-radius: 50%;

overflow: hidden; // Makes it a perfect circle

// Placeholder image inside the avatar

.avatar-placeholder {

width: 100%; // Takes full width of the parent

height: 100%;

background-color: #888;

}

}

}

}

//Container for the main content of the dashboard

// Section with light green background for main content area

.green-container {

background-color: #c1e1c1; // Light green background

border-radius: 16px;//rounded corners

padding: 20px;// Space inside the container

margin-bottom: 20px;// Space below the container to separate it from other elements

flex-grow: 1;//// Allows the container to grow and fill available space in Flexbox

}

// This is the main content area of the dashboard

.dashboard-content {

height: 100%;

}

// This is the container for the cards in the dashboard

.loading-container, .error-container {

display: flex;

justify-content: center;

align-items: center;

min-height: 200px;//// Minimum height so it doesn't collapse while loading

}

/\* Top-left card \*/

:host ::ng-deep .top-left-card .stock-card {

height: 360px;// Fixed height for consistency

border: 2px solid black;

border-radius: 8px; // Slightly rounded corners

background-color: rgb(43, 88, 25);

}

:host ::ng-deep .bottom-left-card .stock-card {

height: 260px; /\* Increase this value as needed \*/

border: 2px solid black;

position: absolute;

width: 97%;

bottom: 0;

background-color: rgb(66, 148, 88); /\* This anchors the card to the bottom \*/

}

/\* Make the container have relative positioning \*/

.col-md-6 {

position: relative;

min-height: 200px; /\* Should match or exceed card height \*/

}

/\* Top-right card (vesting progress) - maintaining original styling \*/

:host ::ng-deep .vesting-card {

height: 200px;

border: 2px solid black;

background-color: rgb(66, 148, 88);

}

/\* Bottom-right card \*/

:host ::ng-deep .bottom-right-card .stock-card {

height: 200px;

border: 2px solid black;

border-radius: 8px;

background-color:rgb(43, 88, 25);

}

/\* Common styling for all cards \*/

:host ::ng-deep .stock-card,

:host ::ng-deep .vesting-card {

border: 2px solid black;

padding: 15px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

margin-bottom: 15px;

}

.footer-label {

width: 100%;

background-color: black;

color: white;

text-align: center;

padding: 10px 0;

font-weight: 500;

font-size: 1rem;

position: fixed;

bottom: 0;

left: 0;

z-index: 999; // Stays above other content

}

/\* Mobile-specific adjustments \*/

@media (max-width: 768px) {

.dashboard-container {

padding: 10px;

}

.green-container {

padding: 15px;

}

.header {

display: flex;

flex-direction: column;

align-items: flex-start;

padding-top: 20px;

}

.title h1 {

font-size: 1.5rem;

}

.top-row {

display: flex;

justify-content: space-between;

align-items: center;

width: 100%;

}

.stock-info {

margin-top: 50px;/\* Increase this value to move stock info further down \*/

font-size: 0.9rem;

width: 100%;

order: 2;

display: flex; /\* Use flexbox inside stock-info \*/

justify-content: center; /\* Centers the stock info horizontally \*/

align-items: center;

.stock-price {

font-size: 0.8rem;

order: 2;

}

}

/\* Keep ESOP title and user info on the same line \*/

.title, .user-info {

display: flex !important;

align-items: center;

justify-content: space-between;

margin-bottom: 1px;

}

.row.g-4 {

display: flex;

flex-wrap: wrap;

margin: 0 -8px;

}

.col-md-6 {

flex: 0 0 50% !important;

width: 50% !important;

padding: 0 8px !important;

min-height: 180px;

}

/\* Fix card heights \*/

:host ::ng-deep .top-left-card .stock-card,

:host ::ng-deep .bottom-left-card .stock-card,

:host ::ng-deep .bottom-right-card .stock-card,

:host ::ng-deep .vesting-card {

height: auto !important;

min-height: 180px;

width: 100% !important;

position: relative !important;

bottom: auto;

}

/\* Adjust card text sizes \*/

:host ::ng-deep .stock-card .card-title,

:host ::ng-deep .vesting-card .card-title {

font-size: 1.5rem !important;

margin-bottom: 10px !important;

}

:host ::ng-deep .stock-card .card-value {

font-size: 2rem !important;

}

/\* Fix vesting progress chart size \*/

:host ::ng-deep .vesting-card .chart-container {

width: 120px !important;

height: 120px !important;

}

:host ::ng-deep .vesting-card .percentage-indicator .percentage {

font-size: 1.5rem !important;

}

:host ::ng-deep .vesting-card .next-vesting-date {

padding: 8px 20px !important;

}

:host ::ng-deep .vesting-card .next-vesting-date .date-label,

:host ::ng-deep .vesting-card .next-vesting-date .date-value {

font-size: 1rem !important;

}

.dashboard-container {

padding-bottom: 60px; // add extra bottom space

}

}

Esop-dashboard.component.ts

// esop-dashboard.component.ts

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

import { EquityService, EquityData } from '../../services/equity.service';

@Component({

selector: 'app-esop-dashboard',

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

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

})

export class EsopDashboardComponent implements OnInit {

equityData: any;

loading = true;

error = false;

constructor(private equityService: EquityService) {}

ngOnInit(): void {

this.loadData();

}

loadData(): void {

this.equityService.getEsopData().subscribe({

next: (data) => {

this.equityData = data;

this.loading = false;

},

error: (err) => {

console.error('Error loading equity data:', err);

this.error = true;

this.loading = false;

}

});

}

}

Esop-vesting-progress.component.html

<!-- esop-vesting-progress.component.html -->

<div class="vesting-card">

<div class="card-title">Vesting Progress</div>

<div class="chart-container">

<canvas baseChart

[datasets]="doughnutChartDatasets"

[labels]="doughnutChartLabels"

[options]="doughnutChartOptions"

[type]="'doughnut'">

</canvas>

<div class="percentage-indicator">

<div class="percentage">{{percentage}}%</div>

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

</div>

</div>

<div class="next-vesting-date">

<div class="date-label">{{nextDate}}</div>

<div class="date-value">${{amountValue}}</div>

</div>

</div>

Esop-vesting-progress.component.scss

// esop-vesting-progress.component.scss

.vesting-card {

background-color: #135e13; // Dark green background

border-radius: 12px;

padding: 25px;

color: white;

height: 100%;

display: flex;

flex-direction: column;

align-items: center;

.card-title {

font-size: 2.4rem;

margin-bottom: 15px;

font-weight: 500;

text-align: center;

}

.chart-container {

position: relative;

width: 180px;

height: 180px;

margin: auto;

@media (max-width: 768px) {

width: 150px;

height: 150px;

}

.chartjs-render-monitor {

/\* Adjusting background color for the inner circle \*/

background-color: #c1e1c1 !important;

}

}

.percentage-indicator {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

text-align: center;

.percentage {

font-size: 2rem;

font-weight: bold;

color: #a5d6a7; // Light green color

}

.label {

font-size: 1rem;

color: #a5d6a7; // Light green color

}

}

.next-vesting-date {

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

border-radius: 8px;

padding: 12px 40px;

text-align: center;

margin-top: 10px;

.date-label {

font-size: 1.2rem;

margin-bottom: 5px;

}

.date-value {

font-size: 1.2rem;

font-weight: bold;

}

}

}

Esop-vesting-progress.component.ts

// esop-vesting-progress.component.ts

import { Component, Input, OnChanges, SimpleChanges } from '@angular/core';

import { ChartConfiguration } from 'chart.js';

@Component({

selector: 'app-esop-vesting-progress',

templateUrl: './esop-vesting-progress.component.html',

styleUrls: ['./esop-vesting-progress.component.scss']

})

export class EsopVestingProgressComponent implements OnChanges {

@Input() percentage: number = 0;

@Input() nextDate: string = '';

@Input() amountValue: number = 0;

public doughnutChartOptions: ChartConfiguration<'doughnut'>['options'] = {

responsive: true,

cutout: '75%',

plugins: {

legend: {

display: false

},

tooltip: {

enabled: false

}

}

};

public doughnutChartLabels: string[] = ['Vested', 'Unvested'];

public doughnutChartDatasets: ChartConfiguration<'doughnut'>['data']['datasets'] = [

{

data: [25, 75],

backgroundColor: ['#135e13', '#a5d6a7'], // Dark green and light green

borderWidth: 0

}

];

ngOnChanges(changes: SimpleChanges): void {

if (changes['percentage']) {

this.updateChartData();

}

}

updateChartData(): void {

const remaining = 100 - this.percentage;

this.doughnutChartDatasets = [

{

data: [this.percentage, remaining],

backgroundColor: ['#135e13', '#a5d6a7'], // Dark green and light green

borderWidth: 0

}

];

}

}

Side-nav.component.html

<!--main container for the side navigation bar-->

<div class="sidenav h-100">

<!-- logo section -->

<div class="logo-container">

<div class="logo">

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

</div>

</div>

<!-- navigation buttons

[class.active]="...": dynamically applies the active class to the selected tab

(click)="...": when clicked, calls setActiveTab() with the tab name.-->

<div class="nav-links">

<button class="nav-btn" [class.active]="activeTab === 'RSU'" (click)="setActiveTab('RSU')">RSU</button>

<button class="nav-btn" [class.active]="activeTab === 'ESOP'" (click)="setActiveTab('ESOP')">ESOP</button>

<button class="nav-btn" [class.active]="activeTab === 'VESTING\_SCHEDULE'" (click)="setActiveTab('VESTING\_SCHEDULE')">Vest</button>

</div>

<!--Logout section that uses bi-bi-box-arrow-right bootstrap icon-->

<div class="logout-btn">

<i class="bi bi-box-arrow-right"></i>

<span>Logout</span>

</div>

<!--Back button that uses bi-bi-arrow-left bootstrap icon-->

<div class="back-btn">

<i class="bi bi-arrow-left"></i>

</div>

</div>

side-nav.component.scss

.sidenav {

background-color: #f5f5f5; //light gray background

display: flex; //Flxbox for layout-flexible and responsive way

flex-direction: column; //vertical stacking of children

align-items: center; //center align children horizontally

padding: 20px 0; //adds vertical padding to the container:20px at the top and bottom,0 on the left and right.

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); // soft shadow around the sidebar

min-height: 100vh; //100% of the viewport height

}

//This styles the container that holds the logo and text of the logo

.logo-container {

display: flex;

flex-direction: column;//vertical stacking of children

align-items: center; //center align children horizontally

margin-bottom: 40px; //space between logo and nav links

}

//This is the container div that holds the image and text of the logo

.logo {

width: 160px;

height: 160px;

background-color: transparent;// no bg color

border-radius: 10px;//rounded corners

display: flex;//flexbox for layout

justify-content: center;// center wrt left to right

align-items: center; //center wrt top to bottom

margin-bottom: 5px;

}

//This styles the actual image inside the .logo container

.logo img {

width: 180px; //image width

height: 180px; //image height

object-fit: contain; //maintain aspect ratio of the image

}

//This styles the text inside the .logo container

.logo-text {

color: #3e7a6f; //teal text color

font-weight: 500; //medium font weight

font-size: 6rem; //relative too root font size

}

//A vertical list of navigation buttons

.nav-links {

width: 100%; //full width of the container parent

display: flex;

flex-direction: column;

align-items: center;

margin-bottom: auto;//Pushes elements below (like logout/back buttons) to the bottom

}

//This styles the individual navigation buttons inside the .nav-links container

.nav-btn {

width: 90%;//90% of the parent container width

padding: 12px;//Space inside the button

margin: 8px 0;//Space inside the button

border: none;//No border

border-radius: 30px;//Rounded pill-shaped corners

background-color: transparent;

font-weight: 500;//Medium bold text

transition: all 0.3s;//Smooth animation on hover/active

&.active {

background-color: #0275d8;//Blue background for the selected tab

color: white;// White text on active tab

}

&:hover:not(.active) {

background-color: #e0e0e0;// Light grey background when hovered (if not active)

}

}

//Provides a simple back icon button with centered icon

.back-btn {

width: 40px;//small square

height: 40px;

border-radius: 50%;//mske it circle

background-color: #f0f0f0;//light grey bg

display: flex;

justify-content: center;

align-items: center;

cursor: pointer;//show hand pointer on hover

// represent icons inside the button

i {

font-size: 20px;//Icon size

color: #666;//medium grey color

}

}

side-nav.component.ts

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

import { EquityService } from '../../services/equity.service';

import { TabService } from '../../services/tab.service';

@Component({

selector: 'app-side-nav',

templateUrl: './side-nav.component.html',

styleUrls: ['./side-nav.component.scss']

})

export class SideNavComponent implements OnInit {

activeTab: string = 'RSU'; // Set default active tab to 'RSU'

username: string = '';

constructor(

private equityService: EquityService,

private tabService: TabService

) {}

ngOnInit(): void {

// Fetch user data when the component is loaded

this.equityService.getEquityData().subscribe({

next: (data) => {

this.username = data.username;

},

error: (err) => {

console.error('Error loading user data:', err);

}

});

// Subscribe to tab changes

this.tabService.activeTab$.subscribe(tab => {

this.activeTab = tab;

});

}

openNav() {

// Placeholder method for opening nav (currently unused)

throw new Error('Method not implemented.');

}

setActiveTab(tab: string) {

// Update which tab is active when a nav button is clicked

this.activeTab = tab;

this.tabService.setActiveTab(tab);

}

}

stock-card.component.html

<div class="stock-card" [ngClass]="colorClass">

<div class="card-content">

<div class="card-title">{{title}}</div>

<div class="card-value">

<span \*ngIf="isCurrency">$</span>{{value}}

</div>

</div>

</div>

stock-card.component.scss

.stock-card {

border-radius: 12px;

padding: 30px;

color: white;

height: 100%;

transition: transform 0.2s;

border-radius: 12px;

flex: 1;

min-width: 250px;

max-width: 100%;

@media (max-width: 768px) {

width: 90%;

margin: auto;

}

&:hover {

transform: translateY(-5px);

}

&.dark-blue {

background-color: #0b476b;

}

&.teal {

background-color: #157a76;

}

.card-content {

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

text-align: center;

}

.card-title {

font-size: 2.4rem;

margin-bottom: 20px;

font-weight: 500;

}

.card-value {

font-size: 3.5rem;

font-weight: bold;

}

}

stock-card.component.ts

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

@Component({

selector: 'app-stock-card',

templateUrl: './stock-card.component.html',

styleUrls: ['./stock-card.component.scss']

})

export class StockCardComponent {

@Input() title: string = '';

@Input() value: number = 0;

@Input() isCurrency: boolean = false;

@Input() colorClass: 'dark-blue' | 'teal' = 'dark-blue';

@Input() previousStockPrice!: number;

Math: any;

}

vesting-progress.component.html

<div class="vesting-card">

<div class="card-title">Vesting Progress</div>

<div class="chart-container">

<canvas baseChart

[datasets]="doughnutChartDatasets"

[labels]="doughnutChartLabels"

[options]="doughnutChartOptions"

[type]="'doughnut'">

</canvas>

<div class="percentage-indicator">

<div class="percentage">{{percentage}}%</div>

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

</div>

</div>

<div class="next-vesting-date">

<div class="date-label">Next Vesting Date</div>

<div class="date-value">{{nextDate}}</div>

</div>

</div>

vesting-progress.component.scss

.vesting-card {

background-color: #157a76;

border-radius: 12px;

padding: 25px;

color: white;

height: 100%;

display: flex;

flex-direction: column;

align-items: center;

.card-title {

font-size: 2.4rem;

margin-bottom: 15px;

font-weight: 500;

text-align: center;

}

.chart-container {

position: relative;

width: 180px;

height: 180px;

margin: auto;

@media (max-width: 768px) {

width: 150px;

height: 150px;

}

.chartjs-render-monitor {

/\* Adjusting background color for the inner circle \*/

background-color: #8fd4c4 !important;

}

}

.percentage-indicator {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

text-align: center;

.percentage {

font-size: 2rem;

font-weight: bold;

color: #0b476b;

}

.label {

font-size: 1rem;

color: #0b476b;

}

}

.next-vesting-date {

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

border-radius: 8px;

padding: 12px 40px;

text-align: center;

margin-top: 10px;

.date-label {

font-size: 1.2rem;

margin-bottom: 5px;

}

.date-value {

font-size: 1.2rem;

font-weight: bold;

}

}

}

vesting-progress.component.ts

import { Component, Input, OnChanges, SimpleChanges } from '@angular/core';

import { ChartConfiguration } from 'chart.js';

@Component({

selector: 'app-vesting-progress',

templateUrl: './vesting-progress.component.html',

styleUrls: ['./vesting-progress.component.scss']

})

export class VestingProgressComponent implements OnChanges {

@Input() percentage: number = 0;

@Input() nextDate: string = '';

public doughnutChartOptions: ChartConfiguration<'doughnut'>['options'] = {

responsive: true,

cutout: '75%',

plugins: {

legend: {

display: false

},

tooltip: {

enabled: false

}

}

};

public doughnutChartLabels: string[] = ['Vested', 'Unvested'];

public doughnutChartDatasets: ChartConfiguration<'doughnut'>['data']['datasets'] = [

{

data: [25, 75],

backgroundColor: ['#0b476b', '#157a76'],

borderWidth: 0

}

];

ngOnChanges(changes: SimpleChanges): void {

if (changes['percentage']) {

this.updateChartData();

}

}

updateChartData(): void {

const remaining = 100 - this.percentage;

this.doughnutChartDatasets = [

{

data: [this.percentage, remaining],

backgroundColor: [ '#0b476b', 'white'], // Inner circle teal, outer ring dark cyan blue

borderWidth: 0

}

];

}

}

vesting-schedule.component.html

<!-- vesting-schedule.component.html -->

<div class="vesting-schedule-container">

<div class="header">

<h1>VESTING SCHEDULE</h1>

</div>

<div class="table-container">

<table class="vesting-table">

<thead>

<tr>

<th [appTooltip]="tooltips.slNo">Sl.No</th>

<th [appTooltip]="tooltips.symbol">Symbol</th>

<th [appTooltip]="tooltips.type">Type</th>

<th [appTooltip]="tooltips.awardPrice">Award Price</th>

<th [appTooltip]="tooltips.granted">Granted</th>

<th [appTooltip]="tooltips.exercisable">Exercisable</th>

<th [appTooltip]="tooltips.finalExerciseDate">Final Exercise Date</th>

<th [appTooltip]="tooltips.unvestedMarketValue">Unvested Market Value</th>

<th [appTooltip]="tooltips.exercisableMarketValue">Exercisable Market Value</th>

<th>Display</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let item of vestingData; let i = index" [ngClass]="{'alternate-row': i % 2 !== 0}">

<td>{{ item.slNo }}</td>

<td>{{ item.symbol }}</td>

<td>{{ item.type }}</td>

<td>{{ item.awardPrice }}</td>

<td>{{ item.granted }}</td>

<td>{{ item.exercisable }}</td>

<td>{{ item.finalExerciseDate }}</td>

<td>{{ item.unvestedMarketValue }}</td>

<td>{{ item.exercisableMarketValue }}</td>

<td>

<button class="view-button" (click)="navigateTo(item.type)">View</button>

</td>

</tr>

</tbody>

</table>

</div>

</div>

<div class="footer-label">

© 2025 EquiHub. All Rights Reserved.

</div>

vesting-schedule.component.scss

/\* vesting-schedule.component.scss \*/

.vesting-schedule-container {

background-color: #f5f9fa; /\* Very light teal/blue-green background \*/

min-height: 100vh;

padding: 20px;

display: flex;

flex-direction: column;

gap: 20px;

}

/\* vesting-schedule.component.scss \*/

.vesting-table {

width: 100%;

border-collapse: collapse;

th, td {

padding: 12px 15px;

text-align: left;

position: relative;

font-weight: bold; /\* Make all table text bold \*/

}

}

.header {

display: flex;

justify-content: flex-start;

align-items: center;

padding-bottom: 20px;

h1 {

color: #00796b; /\* Medium-dark teal for title \*/

font-weight: bold;

margin: 0;

font-size: 2.5rem;

}

}

.table-container {

width: 100%;

overflow-x: auto;

border-radius: 8px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

.vesting-table {

width: 100%;

border-collapse: collapse;

th, td {

padding: 12px 15px;

text-align: left;

position: relative;

}

th {

background-color: #00796b; /\* Medium-dark teal header \*/

color: white;

font-weight: 500;

cursor: help;

}

tbody tr {

background-color: white;

border-bottom: 1px solid #f0f0f0;

&.alternate-row {

background-color: #e0f2f1; /\* Light teal for alternating rows \*/

}

&:hover {

background-color: #f5f5f5;

&.alternate-row {

background-color: #b2dfdb; /\* Slightly darker teal on hover \*/

}

}

}

}

.view-button {

background-color: #00796b; /\* Medium-dark teal button \*/

color: white;

border: none;

border-radius: 4px;

padding: 6px 12px;

cursor: pointer;

font-weight: 500;

transition: background-color 0.3s;

&:hover {

background-color: #004d40; /\* Darker teal on hover \*/

}

}

/\* For tooltip implementation \*/

.tooltip {

position: fixed;

background-color: #c25222; /\* Medium-dark teal tooltip \*/

color: white;

padding: 8px 12px;

border-radius: 4px;

z-index: 100;

max-width: 250px;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

pointer-events: none;

opacity: 0;

transition: opacity 0.3s ease;

}

.tooltip-visible {

opacity: 1;

}

.footer-label {

width: 100%;

background-color: black;

color: white;

text-align: center;

padding: 10px 0;

font-weight: 500;

font-size: 1rem;

position: fixed;

bottom: 0;

left: 0;

z-index: 999; // Stays above other content

}

vesting-schedule.component.ts

// vesting-schedule.component.ts

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

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

import { TabService } from '../../services/tab.service';

interface VestingItem {

slNo: number;

symbol: string;

type: string;

awardPrice: number;

granted: number;

exercisable: number;

finalExerciseDate: string;

unvestedMarketValue: number;

exercisableMarketValue: number;

}

@Component({

selector: 'app-vesting-schedule',

templateUrl: './vesting-schedule.component.html',

styleUrls: ['./vesting-schedule.component.scss']

})

export class VestingScheduleComponent implements OnInit {

vestingData: VestingItem[] = [

{

slNo: 1,

symbol: 'IBM',

type: 'RSU',

awardPrice: 10,

granted: 100,

exercisable: 25,

finalExerciseDate: '21/06/2025',

unvestedMarketValue: 75,

exercisableMarketValue: 250

},

{

slNo: 2,

symbol: 'IBM',

type: 'RSU',

awardPrice: 10,

granted: 100,

exercisable: 6,

finalExerciseDate: '21/06/2026',

unvestedMarketValue: 69,

exercisableMarketValue: 60

},

{

slNo: 3,

symbol: 'IBM',

type: 'ESOP',

awardPrice: 10,

granted: 100,

exercisable: 6,

finalExerciseDate: '21/06/2027',

unvestedMarketValue: 63,

exercisableMarketValue: 60

},

{

slNo: 4,

symbol: 'IBM',

type: 'ESOP',

awardPrice: 10,

granted: 100,

exercisable: 6,

finalExerciseDate: '21/06/2028',

unvestedMarketValue: 57,

exercisableMarketValue: 60

},

{

slNo: 5,

symbol: 'IBM',

type: 'ESOP',

awardPrice: 10,

granted: 100,

exercisable: 7,

finalExerciseDate: '21/06/2029',

unvestedMarketValue: 50,

exercisableMarketValue: 70

},

{

slNo: 6,

symbol: 'IBM',

type: 'RSU',

awardPrice: 12,

granted: 120,

exercisable: 40,

finalExerciseDate: '01/01/2026',

unvestedMarketValue: 80,

exercisableMarketValue: 300

},

{

slNo: 7,

symbol: 'IBM',

type: 'RSU',

awardPrice: 15,

granted: 90,

exercisable: 20,

finalExerciseDate: '15/03/2027',

unvestedMarketValue: 70,

exercisableMarketValue: 200

},

{

slNo: 8,

symbol: 'IBM',

type: 'ESOP',

awardPrice: 9,

granted: 110,

exercisable: 10,

finalExerciseDate: '11/11/2028',

unvestedMarketValue: 60,

exercisableMarketValue: 90

},

{

slNo: 9,

symbol: 'IBM',

type: 'RSU',

awardPrice: 13,

granted: 80,

exercisable: 15,

finalExerciseDate: '30/09/2026',

unvestedMarketValue: 55,

exercisableMarketValue: 150

},

{

slNo: 10,

symbol: 'IBM',

type: 'ESOP',

awardPrice: 11,

granted: 100,

exercisable: 8,

finalExerciseDate: '10/10/2029',

unvestedMarketValue: 65,

exercisableMarketValue: 95

}

];

tooltips = {

slNo: 'Serial Number of the vesting item',

symbol: 'Stock symbol identifier',

type: 'Type of equity compensation',

awardPrice: 'The price at which the equity was granted',

granted: 'Number of shares granted',

exercisable: 'Number of shares that can be exercised',

finalExerciseDate: 'Last date to exercise the options',

unvestedMarketValue: 'Current market value of unvested shares',

exercisableMarketValue: 'Current market value of exercisable shares'

};

//tabService: any;

constructor(

private router: Router,

private tabService: TabService

) { }

ngOnInit(): void {

}

navigateTo(type: string): void {

console.log(`Navigating to type: ${type}`); // Add this for debugging

// First set the active tab

this.tabService.setActiveTab(type);

// Add a small delay before navigation to ensure tab is set

setTimeout(() => {

if (type === 'RSU') {

console.log('Navigating to RSU tab');

} else if (type === 'ESOP') {

console.log('Navigating to ESOP tab');

}

}, 100);

}

showTooltip(tooltip: string): void {

console.log(`Showing tooltip: ${tooltip}`);

}

}

Equity.service.ts

// services/equity.service.ts

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

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

import { Observable, of } from 'rxjs';

export interface EquityData {

username: string;

stockPrice: number;

totalStocksGranted: number;

remainingUnvestedStocks: number;

vestingPercentage: number;

nextVestingDate: string;

currentValue: number;

}

@Injectable({

providedIn: 'root'

})

export class EquityService {

// Removed duplicate getEsopData method to resolve the error.

// For Sprint 1, we'll return static data

// For Sprint 2, this will use HttpClient to fetch from JSON files

constructor(private http: HttpClient) {}

getEquityData(): Observable<EquityData> {

// Static data for Sprint 1

const staticData: EquityData = {

username: 'Deepak Kalwal',

stockPrice: 100,

totalStocksGranted: 100,

remainingUnvestedStocks: 75,

vestingPercentage: 45,

nextVestingDate: 'Aug 30,2025',

currentValue: 2500

};

return of(staticData);

// For Sprint 2, uncomment this to fetch from JSON:

// return this.http.get<EquityData>('assets/data/equity.json');

}

getEsopData(): Observable<EquityData> {

// Static data for ESOP

const staticData: EquityData = {

username: 'Deepak Kalwal',

stockPrice: 100,

totalStocksGranted: 100,

remainingUnvestedStocks: 75,

vestingPercentage: 25,

nextVestingDate: 'Amount Needed to Exercise',

currentValue: 2000

};

return of(staticData);

}

}

Tab.service.ts

// services/tab.service.ts

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

import { BehaviorSubject } from 'rxjs';

@Injectable({

providedIn: 'root'

})

export class TabService {

private activeTabSource = new BehaviorSubject<string>('RSU');

activeTab$ = this.activeTabSource.asObservable();

constructor() {}

setActiveTab(tab: string) {

this.activeTabSource.next(tab);

}

}

app-routing.module.ts

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

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

const routes: Routes = [];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }

App.component.html

<div class="container-fluid p-0">

<div class="row g-0 flex-wrap">

<div class="col-md-2 col-12">

<app-side-nav></app-side-nav>

</div>

<div class="col-md-10 col-12">

<!-- Show different dashboard based on active tab -->

<app-dashboard \*ngIf="activeTab === 'RSU'"></app-dashboard>

<app-esop-dashboard \*ngIf="activeTab === 'ESOP'"></app-esop-dashboard>

<app-vesting-schedule \*ngIf="activeTab === 'VESTING\_SCHEDULE'"></app-vesting-schedule>

</div>

</div>

</div>

App.component.ts

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

import { TabService } from './services/tab.service';

@Component({

selector: 'app-root',

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

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

})

export class AppComponent implements OnInit {

activeTab: string = 'RSU'; // Default to RSU tab

constructor(private tabService: TabService) {}

ngOnInit() {

// Subscribe to tab changes

this.tabService.activeTab$.subscribe(tab => {

this.activeTab = tab;

});

}

}

App.module.ts

// app.module.ts

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

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

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

import { NgChartsModule } from 'ng2-charts';

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

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

import { AppComponent } from './app.component';

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

import { StockCardComponent } from './components/stock-card/stock-card.component';

import { VestingProgressComponent } from './components/vesting-progress/vesting-progress.component';

import { SideNavComponent } from './components/side-nav/side-nav.component';

import { EsopDashboardComponent } from './components/esop-dashboard/esop-dashboard.component';

import { EsopVestingProgressComponent } from './components/esop-vesting-progress/esop-vesting-progress.component';

import { VestingScheduleComponent } from './components/vesting-schedule/vesting-schedule.component';

import { TooltipDirective } from './tooltip.directive';

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

// Ensure the correct path to the VestingScheduleComponent is used

@NgModule({ declarations: [

AppComponent,

DashboardComponent,

StockCardComponent,

VestingProgressComponent,

SideNavComponent,

EsopDashboardComponent,

EsopVestingProgressComponent,

VestingScheduleComponent,

TooltipDirective

],

bootstrap: [AppComponent], imports: [[CommonModule],

BrowserModule,

NgChartsModule], providers: [provideHttpClient(withInterceptorsFromDi())] })

export class AppModule { }

Tooltip.directive.ts

// Updated tooltip.directive.ts

import { Directive, ElementRef, HostListener, Input, Renderer2 } from '@angular/core';

@Directive({

selector: '[appTooltip]'

})

export class TooltipDirective {

@Input('appTooltip') tooltipText: string = '';

tooltip: HTMLElement | null = null;

constructor(private el: ElementRef, private renderer: Renderer2) { }

@HostListener('mouseenter') onMouseEnter() {

if (!this.tooltip && this.tooltipText) {

this.createTooltip();

}

}

@HostListener('mouseleave') onMouseLeave() {

if (this.tooltip) {

this.hideTooltip();

}

}

createTooltip() {

// Create tooltip element

this.tooltip = this.renderer.createElement('div');

this.renderer.addClass(this.tooltip, 'tooltip');

// Add the text

const text = this.renderer.createText(this.tooltipText);

this.renderer.appendChild(this.tooltip, text);

// Add to DOM to get dimensions

this.renderer.appendChild(document.body, this.tooltip);

// Set initial position off-screen to not flash

this.renderer.setStyle(this.tooltip, 'position', 'fixed');

this.renderer.setStyle(this.tooltip, 'top', '-1000px');

this.renderer.setStyle(this.tooltip, 'left', '-1000px');

// Position after a brief delay to ensure tooltip has been rendered

setTimeout(() => this.positionTooltip(), 0);

}

positionTooltip() {

if (!this.tooltip) return;

const hostPos = this.el.nativeElement.getBoundingClientRect();

const tooltipPos = this.tooltip.getBoundingClientRect();

// Calculate position - above the element

const top = hostPos.top - tooltipPos.height - 10;

const left = hostPos.left + (hostPos.width - tooltipPos.width) / 2;

// Check if tooltip would go off screen

const adjustedTop = Math.max(10, top); // Keep at least 10px from top

const adjustedLeft = Math.max(10, Math.min(left, window.innerWidth - tooltipPos.width - 10));

// Set position

this.renderer.setStyle(this.tooltip, 'top', `${adjustedTop}px`);

this.renderer.setStyle(this.tooltip, 'left', `${adjustedLeft}px`);

// Add a visible class for fade-in effect

this.renderer.addClass(this.tooltip, 'tooltip-visible');

}

hideTooltip() {

if (this.tooltip) {

this.renderer.removeChild(document.body, this.tooltip);

this.tooltip = null;

}

}

}