Angular Plants Management App

(Workflow and Concepts Overview)

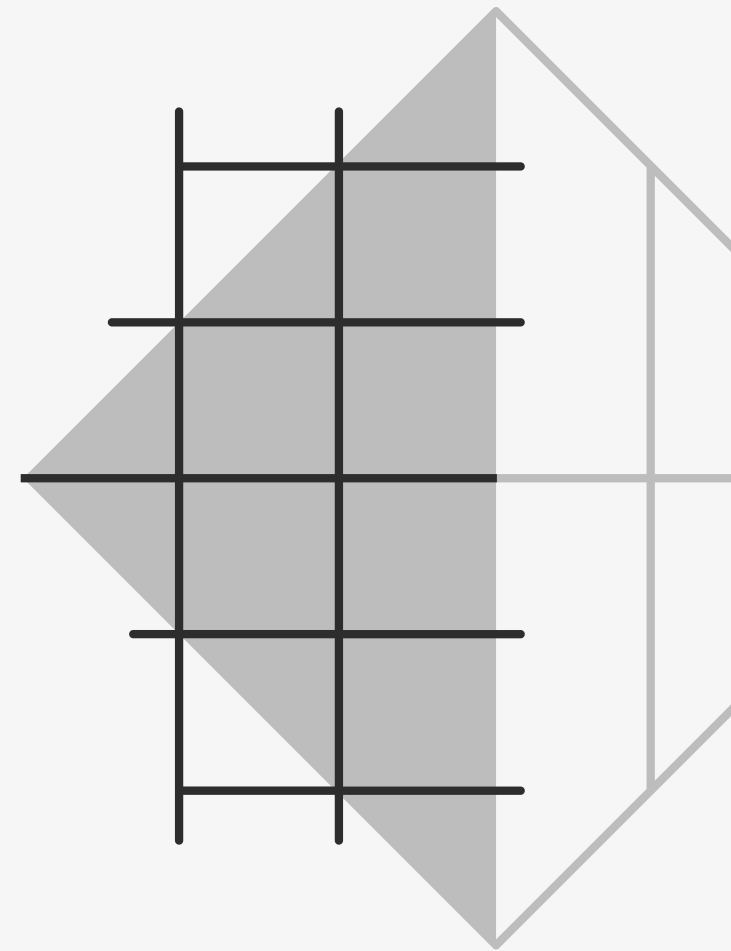31/03/2025   Swathi V

Introduction

A web application that allows users to:

- View a list of plants
- Search for plants
- View detailed plant information
- Mark plants as favorites
- Access favorite plants (Login required)

Technologies Used:

- Angular
- TypeScript and Bootstrap
- Local Storage
- API Calls (JSON file)

App Workflow

How does the app work?

- User visits the Plants List Page (/plants).
- Search plants using the search bar (custom pipe).
- Click "View Details" to see more info (/plants/:id).
- Click "Add to Favorites", which stores the plant in localStorage.
- Visit Favorites Page (/favorites) → Protected by AuthGuard.
- Click "Login" to access the Favorites page.
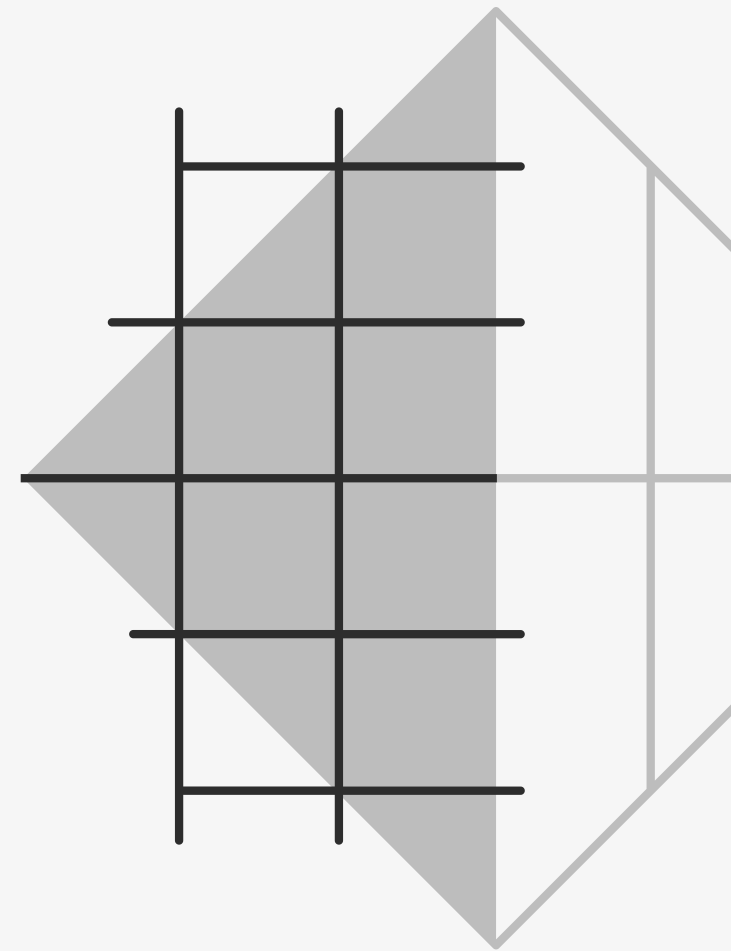
Angular Concepts - Components & Routing

Parent–Child Components
- PlantListComponent (Parent): Displays all plants.
- PlantCardComponent (Child): Displays individual plant details.
- Uses @Input() to pass plant data from parent to child.
- Uses @Output() to send events (favorite plant toggle) from child to parent.

Routing & Parameters
- Dynamic Routing (/plants/:id) → Fetches plant details using ActivatedRoute.
- Navigates between pages using RouterLink.
- Routing Guard (AuthGuard) → Protects Favorites Page by redirecting unauthorized users to the login page.
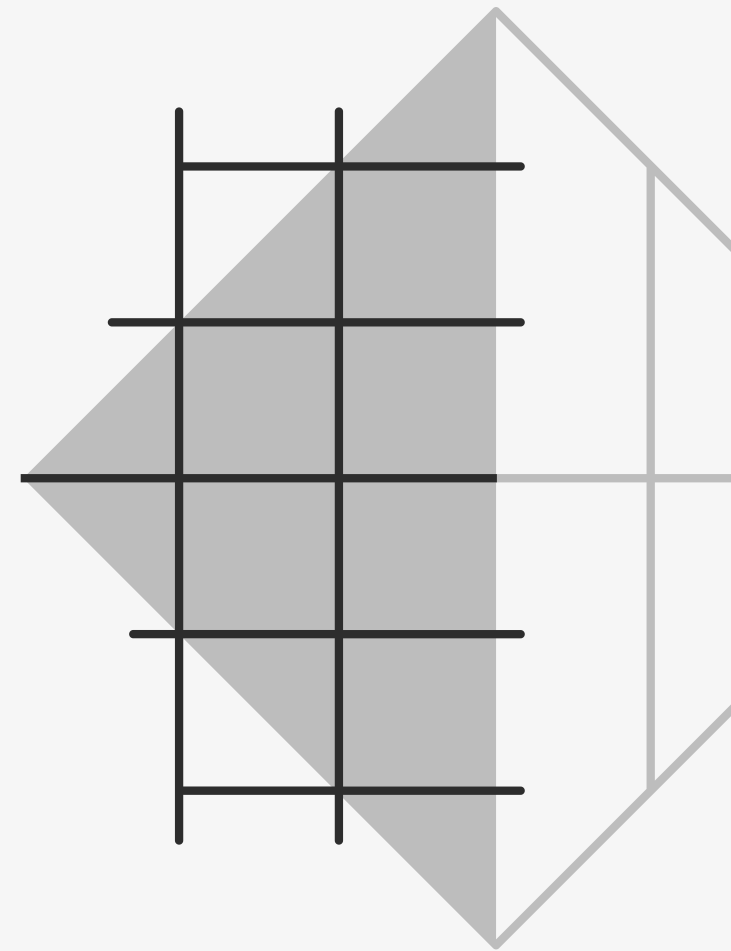
Angular Concepts - Data Handling & Features

Local Storage

- Stores favorite plant IDs persistently.
- Saves login status for authentication.

Services & Observables

- PlantService handles API calls.
- Uses Observables (subscribe()) to fetch plant data asynchronously.
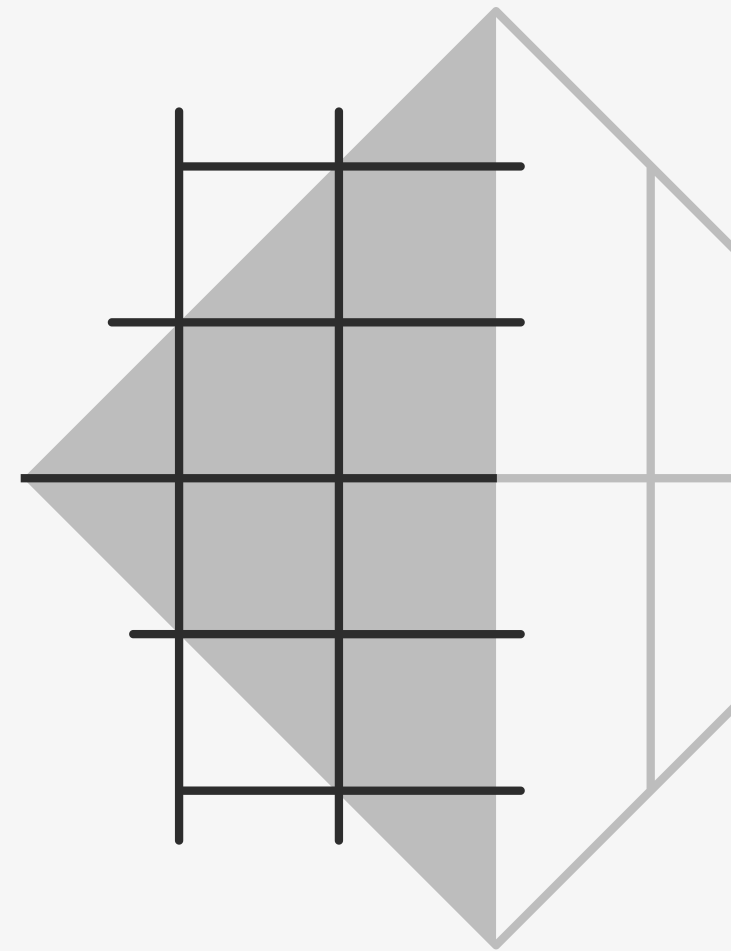
API Call

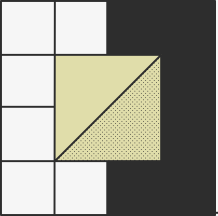- Fetches plant data from api.json file using HttpClient.

Search Feature (Custom Pipe)

- Custom Pipe: PlantFilterPipe
- Filters plants dynamically based on search term.

Form Handling (Login Form)

- Template-Driven Form: Used for Login Form input fields.
- Model-Based (Reactive) Form: Used for Enquiry Form.

Thank you!