1. Component Interaction with @Input and @Output

Create a ProductCatalogComponent that:

Passes a list of products to a ProductItemComponent using @Input.

Receives a notification from ProductItemComponent using @Output when a product is added to the cart.

Gokul

2. Dynamic Forms with Reactive Forms

Build a SurveyFormComponent that:

Dynamically adds form fields based on user input (e.g., number of questions).

Uses Reactive Forms to validate each field as required.

Omkar

3. Custom Directive with Input Property

Create a custom directive called appToggleVisibility that:

Shows or hides an element based on a Boolean input property.

Accepts a delay value to control visibility timing.

Prateek

4. Custom Pipe with Multiple Parameters

Develop a custom pipe named CurrencyConverterPipe that:

Converts an amount from one currency to another.

Takes exchange rate and target currency as parameters.

Anish

5. Two-Way Data Binding with Complex Objects

Implement a ProfileEditorComponent that:

Allows users to update their profile details (name, email, address).

Uses two-way data binding to update nested objects (e.g., address fields).

Dyandev

6. Template Reference Variables with Event Binding

Create a SearchComponent that:

Captures user input using a template reference variable.

Filters a list of items as the user types.

Kartikey

7. Dynamic Component Loading

Build a NotificationService that:

Dynamically loads a NotificationComponent on the screen.

Displays different types of notifications (success, error, warning) using ViewContainerRef.

Sadhya

8. Routing with Child Routes

Develop a UserDashboardComponent that:

Contains child routes for Profile, Settings, and Order History.

Uses <router-outlet> to load the child components.

Ashwina

9. Route Guards with CanDeactivate

Create a CanDeactivateGuard for a EditProfileComponent that:

Prompts the user to save changes before navigating away.

Cancels navigation if the user chooses to stay on the page.

Ayush

10. Dynamic Class Binding and Style Binding

Build a ProductGalleryComponent that:

Applies dynamic classes based on product availability (in-stock, out-of-stock).

Uses style binding to change the opacity for out-of-stock products.

Parth

11. Service for State Management

Develop a CartService that:

Manages the state of the shopping cart.

Shares state between ProductListComponent and CartSummaryComponent.

Suyash

12. Custom Validator for Reactive Forms

Create a custom validator named passwordStrengthValidator that:

Ensures the password contains at least one uppercase letter, one number, and one special character.

Shows dynamic error messages in the form.

Shri nivas

13. Angular Modules with Shared Module

Organize the application by:

Creating a SharedModule for common components, directives, and pipes.

Importing the SharedModule in multiple feature modules.

Prachi

14. Parent-Child Communication Using ViewChild

Build a TimerComponent that:

Uses @ViewChild() to control start, pause, and reset actions from a parent component.

Displays the elapsed time in real-time.

Rutuja

15. Nested Component Communication using @Input and @Output

Objective:

Create a parent-child component setup where:

The parent (OrderListComponent) passes a list of orders to the child (OrderItemComponent).

The child component notifies the parent when an order is marked as shipped.

Ashwarya

16. Dynamic Routing with Route Guards

Objective:

Create a routing system that:

Redirects unauthenticated users to the login page.

Shivani

17. State Management using BehaviorSubject

Objective:

Build a shopping cart system using BehaviorSubject for state management.

A ProductListComponent displays products with an "Add to Cart" button.

A CartComponent displays the cart items and the total amount.

The state of the cart is managed using a shared service with BehaviorSubject.

Abid

18. Dynamic Dropdown with Event Binding

Objective:

Create a CategorySelectorComponent that:

Displays a dropdown of categories.

Dynamically updates a list of products based on the selected category.

Lavanya

19. Reactive Forms with Validation

Create a RegistrationFormComponent that:

Uses Reactive Forms to handle form controls.

Validates that the password is at least 8 characters and the email is in a valid format.

Anjali