









Angular 18 Practical Assignment (PART – II)

1. Use `@Input()` to pass user data to child component
 *Scenario:* A parent dashboard passes user profile data to a child component showing profile summary.
2. Emit data from child to parent using `@Output()`
 *Scenario:* A child rating component sends the selected star rating to the parent product component.
3. Create a simple login form with validation
 *Scenario:* An e-commerce site requires email/password validation for user login.
4. Add a route with parameters like `/product/:id`
 *Scenario:* Clicking a product takes users to `/product/123` to see the product details.
5. Fetch parameter value using `ActivatedRoute`
 *Scenario:* Use the product ID in the URL to fetch details from an API.
6. Load user data from a mock JSON file
 *Scenario:* Display a user list from a local JSON file during development.
7. Display API data using `HttpClient`
 *Scenario:* Fetch and show blog posts from a REST API.
8. Handle API error and show user-friendly message
 *Scenario:* Show an alert if the API fails when loading user data.

9. Create a shared module for reusable components

📖 *Scenario:* Centralize commonly used buttons, modals, and cards in one module.

10. Add a loading spinner while fetching data

📖 *Scenario:* Show a spinner while waiting for API results in a dashboard.

11. Use environment.ts for API URLs

📖 *Scenario:* Switch between development and production API URLs using environment files.

12. Create a ProductCard component and reuse it in a product list

📖 *Scenario:* Show products using the same styled card across the site.

13. Apply custom validators in reactive forms

📖 *Scenario:* Validate that a password has a special character and a number.

14. Add form control dynamically in a form

📖 *Scenario:* Add extra phone number fields when the user clicks "Add more."

15. Write a custom pipe to reverse a string

📖 *Scenario:* Display names or codes in reverse for visual effects or debugging.

16. Store user login data in localStorage

📖 *Scenario:* Keep a user logged in even after a page refresh.

17. Implement CanActivate route guard

📖 *Scenario:* Prevent non-logged-in users from accessing the admin dashboard.

18. Create a SidebarComponent and show/hide based on screen size

📖 *Scenario:* A responsive admin panel shows sidebar only on larger screens.

19. Create a tab switcher using ngIf and ngClass

 **Scenario:** Switch between Product Info, Reviews, and Specs on a product page.

20. Add HttpInterceptor for setting auth headers

 **Scenario:** Automatically attach JWT tokens to every outgoing API request.

21. Lazy-load AdminModule using Angular routing

 **Scenario:** Load admin panel code only when admin logs in.

22. Add pagination to a list fetched from API

 **Scenario:** Show 10 users per page with Next/Previous buttons.

23. Sort and filter a list of items using pipes

 **Scenario:** Filter a product list by category and sort by price.

24. Upload an image and preview before submission

 **Scenario:** Allow users to preview their profile picture before uploading.

25. Use FormBuilder to build form dynamically

 **Scenario:** Use FormBuilder to generate a complex job application form quickly.