Welcome to the Apprenticeship Program – Session 34

Agenda

- Angular Material – Overview & Setup
- Using UI Components & Layouts
- Unit & End-to-End Testing
- Jasmine, Karma, Cypress, Protractor
- Building & Optimizing Angular Apps
- Deployment Best Practices
- Recap & Q&A

Angular Material & UI Components

- A UI component library that follows Google's Material Design principles.
- Built and maintained by the Angular team.
- Offers responsive, accessible, and **themeable** components.

✅ Speeds up UI development with consistent styling.

Setting Up Angular Material

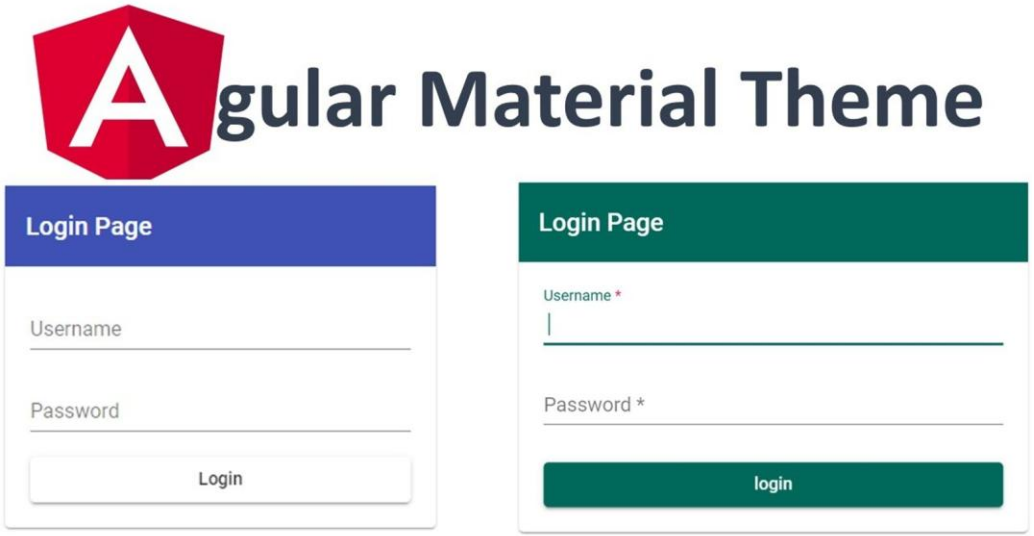
Install it using Angular CLI:

```
ng add @angular/material
```

Select:

- Theme (e.g., Indigo/Pink)
- Global typography
- Animations

- 🎯 This adds Material modules & updates styles automatically.



Common Angular Material Components

Component	Purpose
MatToolbar	Top app bar/header
MatButton	Styled buttons
MatInput	Input fields
MatCard	Cards for layouts
MatTable	Tabular data
MatSnackBar	Toast notifications

```
<mat-form-field>
  <input matInput placeholder="Email">
</mat-form-field>
```

DEMO

🧪 Testing in Angular

- ✅ Ensure features work as expected
- ✅ Catch bugs early
- ✅ Support refactoring and CI/CD pipelines
- ✅ Build user confidence
- Types:
  - Unit Testing – Test individual components/services
  - E2E Testing – Simulate real user interactions

UNIT TESTING – JASMINE & KARMA

- Setup – Jasmine & Karma:
  - ✅ Included by default in Angular CLI
  - ✅ Files auto-generated with .spec.ts
  - ✅ Run tests:

```
ng test
```

This launches Karma test runner in a browser.

Jasmine Basics – Keywords

Keyword	Description
`describe()`	A test suite (group of tests)
`it()`	A single test/spec
`expect()`	Defines the expectation
`beforeEach()`	Runs before each test in the suite
`spyOn()`	Tracks function calls

Unit Test Example – Component

```
describe('GreetingComponent', () => {
  let component: GreetingComponent;

  beforeEach(() => {
    component = new GreetingComponent();
  });

  it('should show the correct greeting', () => {
    component.name = 'Alice';
    expect(component.greet()).toBe('Hello, Alice!');
  });
```

✅ Tests the greet() method behavior

🌐 E2E TESTING – CYPRESS / PROTRACTOR

- Setup – Cypress:

  ✅ Install Cypress:

```
npm install cypress --save-dev
```

  ✅ Add to package.json scripts:

```
"e2e": "cypress open"
```

  ✅ Launch Cypress:

```
npm run e2e
```

Cypress Test Example

```
describe('Login Form', () => {
  it('logs in a user', () => {
    cy.visit('/login');
    cy.get('input[name="email"]').type('user@example.com');
    cy.get('input[name="password"]').type('123456');
    cy.contains('Login').click();
    cy.url().should('include', '/dashboard');
  });
});
```

✅ Simulates real user input and verifies behavior

Legacy Testing Framework

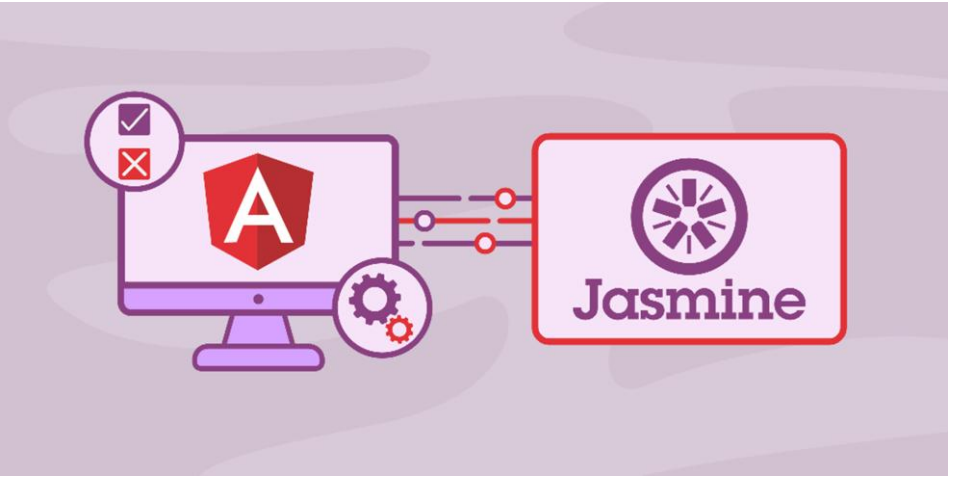
- Protractor Overview:
  - ✅ Older Angular-specific E2E tool
  - ✅ Uses WebDriver (Selenium-based)
  - ✅ Run with:

```
ng e2e
```

- Being deprecated in favor of Cypress or Playwright.

🧪 Testing Best Practices

- ✅ Write one it() block per behavior
- ✅ Avoid testing private methods
- ✅ Use spyOn() to mock services
- ✅ Test components independently of DOM
- ✅ Write E2E tests for critical user flows



DEMO

🚀 Deployment & Optimization

Use Angular CLI to build optimized output:

```
ng build --configuration production
```

💡 This includes:

- Ahead-of-Time (AOT) compilation
- Minification & bundling
- Tree-shaking unused code

Output goes to the /dist folder.

Angular + webpack

Deployment Options

- Common Hosting Platforms:
  - ✅ Firebase Hosting
  - ✅ Netlify / Vercel
  - ✅ AWS S3 + CloudFront
  - ✅ GitHub Pages
  - ✅ Custom server (Node.js, Nginx)

```
npm install -g firebase-tools
firebase init
firebase deploy
```

Performance Optimization Tips

Improve speed and UX by:

- Lazy loading modules
- PreloadingStrategy for anticipated routes
- Image & asset compression
- Use trackBy in *ngFor
- Minimize third-party scripts
- Analyze bundle size with source-map-explorer

DEMO

Conclusion and Q&A

- **Key Takeaways:**
  - Use Angular Material for consistent, responsive Uis
  - Write unit tests with Jasmine & Karma
  - Add end-to-end tests using Cypress or Protractor
  - Use CLI to build optimized bundles
  - Deploy via Firebase, Netlify, or custom setup

Questions? 👀