

Web Development Lab 4: Angular

Submission: Upload your work to a GitHub repository before the next practice lesson

Show Your Understanding of Angular

Learning Objectives

By the end of this lab, you will be able to:

- Understand the Angular project structure and its core building blocks (modules, components, templates)
- Create and configure a new Angular project using the Angular CLI
- Build reusable components with data binding and event handling
- Use `*ngFor` and `*ngIf` structural directives to render dynamic lists and conditional content
- Pass data between components using `@Input()` and `@Output()` decorators
- Implement basic routing and navigation between views
- Integrate external links and sharing functionality into a web application
- Apply clean, component-scoped CSS styling following Angular best practices

Prerequisites

Before starting this lab, ensure you have the following installed on your machine:

- **Node.js** (v18 or later) — <https://nodejs.org/en/>
- **npm** (comes with Node.js) — <https://www.npmjs.com/>
- **Angular CLI** — install globally via: `npm install -g @angular/cli`
- **IDE** — WebStorm (recommended) or VS Code with Angular Language Service extension
- **Git** — for version control and submission

Verify your setup by running `node -v`, `npm -v`, and `ng version` in your terminal.

Tasks (2)

Task 1: Complete the Official Angular Tutorial

Work through the official Angular tutorial to build a foundational understanding of the framework.

- a. Go to <https://angular.dev/tutorials/learn-angular> and complete all steps of the “Learn Angular” tutorial
- b. Follow along by writing the code yourself (do not just read — type every line)
- c. Make sure your tutorial project builds and runs without errors using `ng serve`
- d. Commit and push the completed tutorial project to your GitHub repository

Note: The tutorial covers components, templates, directives, services, routing, and HTTP. Make sure you understand each concept before moving on to Task 2.

Task 2: Build an "Online Store" Application

Create a new Angular project from scratch that displays a catalog of real products from kaspi.kz. This task tests your ability to apply Angular concepts independently.

2.1 Project Setup

- Create a new Angular project: `ng new online-store`
- Use standalone components (Angular 17+ default) or modules — your choice
- Enable routing when prompted during project creation
- Verify the project runs with `ng serve` and opens at `http://localhost:4200`

2.2 Product Interface / Model

Define a TypeScript interface (e.g., `product.model.ts`) with the following properties for each product:

- `id: number` — unique identifier
- `name: string` — product name
- `description: string` — short product description (2–3 sentences)
- `price: number` — product price in KZT
- `rating: number` — rating from 1 to 5 (can be decimal, e.g. 4.7)
- `image: string` — URL or local path to the main product image
- `images: string[]` — array of image URLs for the gallery (minimum 3)
- `link: string` — direct URL to the product page on kaspi.kz

2.3 Products Component

Create a `ProductListComponent` that displays the catalog of products.

- The component must render **at least 10 products** from kaspi.kz
- Each product card must display: image, name, description, price, and rating
- Use `*ngFor` to iterate over the products array
- Display the rating visually (e.g., star icons, filled/empty stars, or a numeric badge)
- Each product card should link to the real kaspi.kz product page (opens in a new tab)
- The layout should be responsive — use CSS Grid or Flexbox for the product grid

2.4 Share Functionality

Each product card must have a “**Share**” button. When clicked, the user should be redirected to WhatsApp or Telegram with a pre-filled message containing the product link.

- Use WhatsApp’s share URL: `https://wa.me/?text=Check out this product: {kaspi_link}`
- Or Telegram’s share URL: `https://t.me/share/url?url={kaspi_link}&text={product_name}`
- You may provide one or both options (bonus for implementing both with a dropdown or icons)
- The share link must include the actual kaspi.kz URL of the specific product

- e. Use `encodeURIComponent()` to safely encode URLs in the share link

2.5 Product Image Gallery (Optional / Bonus)

Implement an image gallery for each product to showcase multiple product images.

- a. Each product should have a minimum of **3 images**
- b. Implement a gallery viewer — e.g., a carousel, lightbox, or thumbnail navigation
- c. Clicking a thumbnail should update the main displayed image
- d. Consider adding previous/next navigation arrows
- e. The gallery can be displayed inline on the product card or in a modal/overlay when clicked

Requirements

- **Angular CLI:** The project must be generated using `ng new` and follow standard Angular project structure
- **Components:** Use at least 2 components (e.g., `ProductListComponent` and `ProductCardComponent`); extract reusable pieces into separate components
- **TypeScript:** Define interfaces for data models; use strong typing throughout (avoid `any`)
- **Data Binding:** Use Angular's property binding `[property]`, event binding `(event)`, and interpolation `{{ }}` appropriately
- **Directives:** Use structural directives (`*ngFor`, `*ngIf`) for dynamic rendering
- **CSS Styling:** Apply clean, scoped component styles; use Flexbox or CSS Grid for layout; the design should be visually polished and consistent
- **Responsiveness:** The product grid should adapt to different screen sizes (desktop, tablet, mobile)
- **Code Quality:** Use meaningful component, variable, and method names; keep components focused and concise; follow the Angular Style Guide
- **Real Data:** All 10+ products must link to actual product pages on kaspi.kz

Deliverables

Submit your GitHub repository containing the following structure:

```
lab4/
  task1/
    (completed Angular tutorial project)
  task2/
    online-store/
      src/
        app/
          models/
            product.model.ts
          components/
            product-list/
              product-list.component.ts
              product-list.component.html
              product-list.component.css
            product-card/
```

```
        product-card.component.ts
        product-card.component.html
        product-card.component.css
    app.component.ts
    app.component.html
    assets/
        images/      (if using local images)
angular.json
package.json
README.md
```

Note: Do not push the `node_modules/` folder. Make sure your `.gitignore` excludes it. Include a `README.md` with instructions to run the project (`npm install` then `ng serve`).

Useful Resources

1. Angular Official Documentation: <https://angular.dev/overview>
2. Angular "Learn Angular" Tutorial: <https://angular.dev/tutorials/learn-angular>
3. Node.js: <https://nodejs.org/en/>
4. npm: <https://www.npmjs.com/>
5. Angular CLI Reference: <https://angular.dev/tools/cli>
6. TypeScript Handbook: <https://www.typescriptlang.org/docs/handbook/>
7. WebStorm IDE: <https://www.jetbrains.com/ru-ru/webstorm/>
8. Angular Style Guide: <https://angular.dev/style-guide>
9. Kaspi.kz (for product data): <https://kaspi.kz/>
10. WhatsApp Share URL Docs: <https://faq.whatsapp.com/5913398998672934>

GOOD LUCK! :)