Developing an Angular Application for Comparing Electricity Tariffs

Objective:

Create a single-page web application for displaying and comparing electricity tariffs. The application should include essential features such as displaying a list of tariffs, sorting, comparing, and more, with responsive design and containerization.

Requirements:

- The application should be created using **the latest stable version of Angular**.
- Use **Angular CLI** for project setup.
- Follow component-based architecture principles.
- Use Angular Services for state management and data handling.
- Implement Angular Reactive Forms for filtering and sorting tariffs.
- Style the application using **SCSS**.
- Provide instructions in the **README** file for building and running the **Docker container** and running **tests**.

If you have any questions about the assignment, feel free to ask for clarifications.

User Story 1: Display Available Tariffs

As a user. I want to see a list of available electricity tariffs on the main page. So that I can browse through different tariff options.

Acceptance Criteria:

- A list of tariffs request from the (**Mock**)
- A list of tariffs is displayed on the main page.
- Each tariff shows key information (name, price, supplier, description).
- All created components, services, etc. should be covered by **unit tests**.

Mock Data:


```
[ {
  "id": 1,
  "name": "Tariff A",
  "price": 3.5,
  "supplier": "Company A",
  "description": "Description of Tariff A"
 }, {
  "id": 2,
  "name": "Tariff B",
  "price": 4.0,
  "supplier": "Company B",
  "description": "Description of Tariff B"
 }, {
  "id": 3,
  "name": "Tariff C",
  "price": 3.8,
  "supplier": "Company C",
  "description": "Description of Tariff C"
 }]
```

User Story 2: Sort Tariffs by Price

As a user. I want to sort tariffs by price in ascending or descending order. So that I can easily find the cheapest or most expensive options.

Acceptance Criteria:

- A control (e.g., a dropdown or buttons) is available to sort tariffs by price.
- Tariffs are sorted correctly in ascending or descending order when the control is used.
- All created components, services, etc. should be covered by unit tests.

User Story 3: Add Tariff to Comparison List

As a user. I want to add tariffs to a comparison list. So that I can compare selected tariffs side-by-side.

Acceptance Criteria:

- Each tariff in the list has an “Add to Compare” button.
- Clicking the “Add to Compare” button adds the tariff to the comparison list.
- The comparison list can hold a maximum of 3 tariffs.
- All created components, services, etc. should be covered by unit tests.

User Story 4: Remove Tariff from Comparison List

As a user. I want to remove tariffs from the comparison list. So that I can update my selection of tariffs to compare.

Acceptance Criteria:

- Each tariff in the comparison list has a “Remove” button.
- Clicking the “Remove” button removes the tariff from the comparison list.
- All created components, services, etc. should be covered by unit tests.

User Story 5: Compare Selected Tariffs

As a user. I want to compare selected tariffs on a separate comparison page. So that I can see the key differences between them.

Acceptance Criteria:

- A comparison page is accessible via a link or button on the main page.
- The comparison page displays all tariffs added to the comparison list.
- Key parameters of the tariffs are displayed in a table format for easy comparison.

- All created components, services, etc. should be covered by unit tests.

User Story 6: Navigate Between Pages

As a user. I want to navigate between the main page and the comparison page. So that I can easily switch between browsing tariffs and comparing them.

Acceptance Criteria:

- The main page is accessible at the path /.
- The comparison page is accessible at the path /compare.
- Navigation is implemented using Angular Router.

User Story 7: Responsive Design

As a user. I want to have a responsive design that works on both mobile and desktop devices. So that I can use the application on any device.

Acceptance Criteria:

- The application displays correctly on mobile devices and desktops.
- The layout adjusts based on the device's screen size (using **media queries**).
- UI elements and controls are user-friendly on all devices.

User Story 8: Containerize the Application

As a developer. I want to containerize the application using Docker. that I can ensure consistent deployment across different environments.

Acceptance Criteria:

- A Dockerfile is provided to build and run the application in a container.
- The application runs correctly inside the Docker container.

Submission Instructions:

1. Create a new local **GIT repository** on your file system.
2. Place your projects source code in this repository.
3. Write a **README** file with instructions on how to run the application, including commands for building and running the Docker container, as well as instructions for running tests.
4. ZIP the repository and send it over via E-Mail.

If you have any questions about the assignment, feel free to ask for clarifications.
