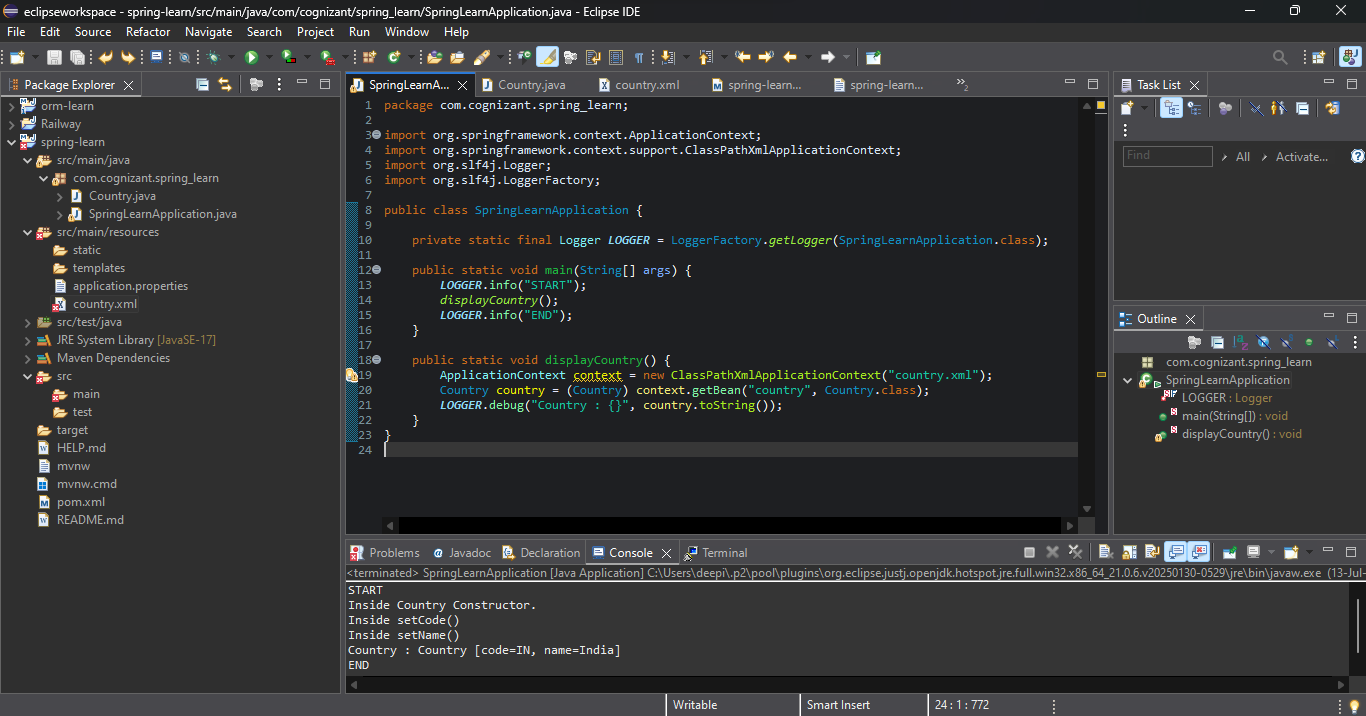
**Screenshots**



# 🛫 Spring XML Country Configuration

## 📄 Description

This project demonstrates the use of \*\*Spring Framework's XML-based configuration\*\* to define and retrieve a bean representing a country, in the context of an \*\*airline booking website\*\*.

The application supports booking from four countries:

- 🇺🇸 United States (`US`)

- 🇩🇪 Germany (`DE`)

- 🇮🇳 India (`IN`)

- 🇯🇵 Japan (`JP`)

Each country is represented by a two-character ISO code and a name. One country (e.g., `India`) is configured in a Spring XML file (`country.xml`) using a `<bean>` definition.

### Key Concepts Covered:

- Defining a bean in XML using `<bean>` and `<property>` tags

- Loading the configuration using `ClassPathXmlApplicationContext`

- Injecting and retrieving beans using Spring's `ApplicationContext`

- Using constructor and method-level logging to trace bean lifecycle

### Flow Summary:

1. A `Country` Java class contains `code` and `name` properties with appropriate getters, setters, and `toString()` method.

2. The class includes debug logs in:

- Constructor (`"Inside Country Constructor."`)

- Getter and setter methods

3. The Spring bean is defined in `country.xml`.

4. `SpringLearnApplication` loads this config and retrieves the bean using `context.getBean("country")`.

5. Bean details are printed using a logger.

This is a foundational Spring exercise designed to teach XML-based dependency injection, bean lifecycle, and basic application context loading.