## Error1

Create a web application that consists of two pages.
The first one allows a user to provide two integer numbers and displays the result of their division.
In case the user provides 0 as the denominator he or she shall be redirected to the second page with a short message saying that this operation is not allowed.
Additional requirements:

- a single user should be able to perform the division operation unlimited number of times (even after providing 0 as the denominator)
- all the pages must be developed using JSF technology (in Jakarta EE 10)
- no persistence or database is required

Struktura

```
MyWebApp/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   └── com/
│   │   │       └── example/
│   │   │           └── mywebapp/
│   │   │               ├── beans/
│   │   │               │   └── DivisionBean.java
│   │   ├── resources/
│   │   │   └── META-INF/
│   │   │       └── faces-config.xml
│   │   ├── webapp/
│   │   │   ├── index.xhtml
│   │   │   ├── error.xhtml
│   │   │   └── WEB-INF/
│   │   │       └── web.xml
└── pom.xml
```

**pom.xml**

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.tmq</groupId>
    <artifactId>Error1</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>war</packaging>
    <name>Error1-1.0-SNAPSHOT</name>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <jakartaee>10.0.0</jakartaee>
    </properties>

    <dependencies>
        <dependency>
            <groupId>jakarta.platform</groupId>
            <artifactId>jakarta.jakartaee-api</artifactId>
            <version>${jakartaee}</version>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>jakarta.faces</groupId>
            <artifactId>jakarta.faces-api</artifactId>
            <version>4.0.1</version>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.1</version>
                <configuration>
                    <source>11</source>
                    <target>11</target>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-war-plugin</artifactId>
                <version>3.3.2</version>
            </plugin>
        </plugins>
    </build>
</project>
```

**DivisionBean.java**

```
package com.tmq.error1.beans;

import jakarta.enterprise.context.RequestScoped;
import jakarta.inject.Named;
import jakarta.faces.context.FacesContext;
import jakarta.servlet.http.HttpServletRequest;
import java.io.IOException;

@Named
@RequestScoped
public class DivisionBean {
    private int numerator;
    private int denominator;
    private String result;

    public int getNumerator() {
        return numerator;
    }

    public void setNumerator(int numerator) {
        this.numerator = numerator;
    }

    public int getDenominator() {
        return denominator;
    }

    public void setDenominator(int denominator) {
        this.denominator = denominator;
    }

    public String getResult() {
        return result;
    }
    public void setResult(String result) {
        this.result = result;
    }

    public void performDivision() {
        if (denominator == 0) {
            try {

FacesContext.getCurrentInstance().getExternalContext().redirect("error.xhtml");
            } catch (IOException e) {
                e.printStackTrace();
            }
        } else {
            result = String.valueOf((double) numerator / denominator);
        }
    }
}
```

**index.xhtml**

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
    <title>Division Operation</title>
</h:head>
<h:body>
    <h1>Perform Division</h1>
    <h:form>
        <h:outputLabel for="numerator" value="Numerator:"/>
        <h:inputText id="numerator" value="#{divisionBean.numerator}"
required="true" requiredMessage="Numerator is required"/>
        <br/><br/>
        <h:outputLabel for="denominator" value="Denominator:"/>
        <h:inputText id="denominator" value="#{divisionBean.denominator}"
required="true" requiredMessage="Denominator is required"/>
        <br/><br/>
        <h:commandButton value="Divide"
action="#{divisionBean.performDivision}"/>
        <br/><br/>
        <h:outputText value="Result: #{divisionBean.result}" rendered="#{not
empty divisionBean.result}"/>
        <br/><br/>
        <h:message for="numerator" style="color:red;"/>
        <h:message for="denominator" style="color:red;"/>
    </h:form>
</h:body>
</html>
```

**error.xhtml**

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
    <title>Error</title>
</h:head>
<h:body>
    <h1>Error</h1>
    <p>Division by zero is not allowed.</p>
    <h:link outcome="index" value="Try again"/>
</h:body>
</html>
```

**faces-config.xml** (wystarczy utworzyć w resources/META-INF)

```
<?xml version='1.0' encoding='UTF-8'?>
<faces-config version="4.0"
              xmlns="https://jakarta.ee/xml/ns/jakartaee"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-facesconfig_4_0.xsd">

</faces-config>
```

**web.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="6.0" xmlns="https://jakarta.ee/xml/ns/jakartaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd">
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <context-param>
        <param-name>jakarta.faces.CONFIG_FILES</param-name>
        <param-value>/WEB-INF/faces-config1.xml</param-value>
    </context-param>
    <servlet>
        <servlet-name>Faces Servlet</servlet-name>
        <servlet-class>jakarta.faces.webapp.FacesServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Faces Servlet</servlet-name>
        <url-pattern>*.xhtml</url-pattern>
    </servlet-mapping>
    <welcome-file-list>
        <welcome-file>index.xhtml</welcome-file>
    </welcome-file-list>
</web-app>
```

## Display2

create a web application that stores on a server a set of information about different people, such as their name, address, and phone number.

Within the web application there should be a single web page that displays these informations in a table, each row is associated to a single person.

Additional requirements:

- the web page must be developed using JSF technology
- people may be duplicated
- no database is required (!)

Struktura

```
MyWebApp/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   └── com/
│   │   │       └── example/
│   │   │           └── mywebapp/
│   │   │               ├── beans/
│   │   │               │   └── PeopleBean.java
│   │   ├── resources/
│   │   │   └── META-INF/
│   │   │       └── faces-config.xml
│   │   ├── webapp/
│   │   │   ├── index.xhtml
│   │   │   └── WEB-INF/
│   │   │       └── web.xml
└── pom.xml
```

**pom.xml**

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.tmq</groupId>
    <artifactId>Display2</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>war</packaging>
    <name>Display2-1.0-SNAPSHOT</name>



    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <jakartaee>10.0.0</jakartaee>
    </properties>

    <dependencies>
        <dependency>
            <groupId>jakarta.platform</groupId>
            <artifactId>jakarta.jakartaee-api</artifactId>
            <version>${jakartaee}</version>
            <scope>provided</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.1</version>
                <configuration>
                    <source>11</source>
                    <target>11</target>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-war-plugin</artifactId>
                <version>3.3.2</version>
            </plugin>
        </plugins>
    </build>
</project>
```

**PeopleBean.java**

```
package com.tmq.display2.beans;

import jakarta.annotation.PostConstruct;
import jakarta.enterprise.context.ApplicationScoped;
import jakarta.inject.Named;

import java.util.ArrayList;
import java.util.List;

@Named
@ApplicationScoped
public class PeopleBean {
    private List<Person> people;
    private String name;
    private String address;
    private String phoneNumber;

    @PostConstruct
    public void init() {
        people = new ArrayList<>();
    }

    public List<Person> getPeople() {
        return people;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public void addPerson() {
        people.add(new Person(name, address, phoneNumber));
        // Clear input fields after adding
        name = "";
        address = "";
        phoneNumber = "";
    }

    public static class Person {
        private String name;
        private String address;
        private String phoneNumber;

        public Person(String name, String address, String phoneNumber) {
            this.name = name;
            this.address = address;
            this.phoneNumber = phoneNumber;
        }

        public String getName() {
            return name;
        }

        public String getAddress() {
            return address;
        }

        public String getPhoneNumber() {
            return phoneNumber;
        }
    }
}
```

**index.xhtml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/XHtml.xhtml to edit
this template
-->
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:f="http://xmlns.jcp.org/jsf/core">
<h:head>
    <title>People Information</title>
</h:head>
<h:body>
    <h1>Add Person</h1>
    <h:form>
```

```
        <h:outputLabel for="name" value="Name:"/>
        <h:inputText id="name" value="#{peopleBean.name}" required="true"
requiredMessage="Name is required"/>
        <br/><br/>
        <h:outputLabel for="address" value="Address:"/>
        <h:inputText id="address" value="#{peopleBean.address}" required="true"
requiredMessage="Address is required"/>
        <br/><br/>
        <h:outputLabel for="phoneNumber" value="Phone Number:"/>
        <h:inputText id="phoneNumber" value="#{peopleBean.phoneNumber}"
required="true" requiredMessage="Phone Number is required"/>
        <br/><br/>
        <h:commandButton value="Add Person" action="#{peopleBean.addPerson}"/>
        <br/><br/>
        <h:message for="name" style="color:red;"/>
        <h:message for="address" style="color:red;"/>
        <h:message for="phoneNumber" style="color:red;"/>
    </h:form>

    <h1>People List</h1>
    <h:dataTable value="#{peopleBean.people}" var="person" border="1">
        <h:column>
            <f:facet name="header">Name</f:facet>
            #{person.name}
        </h:column>
        <h:column>
            <f:facet name="header">Address</f:facet>
            #{person.address}
        </h:column>
        <h:column>
            <f:facet name="header">Phone Number</f:facet>
            #{person.phoneNumber}
        </h:column>
    </h:dataTable>
</h:body>
</html>
```

**web.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="6.0" xmlns="https://jakarta.ee/xml/ns/jakartaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd">
    <context-param>
        <param-name>jakarta.faces.PROJECT_STAGE</param-name>
        <param-value>Development</param-value>
    </context-param>
    <servlet>
        <servlet-name>Faces Servlet</servlet-name>
        <servlet-class>jakarta.faces.webapp.FacesServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Faces Servlet</servlet-name>
        <url-pattern>*.xhtml</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
```

```
        <welcome-file>index.xhtml</welcome-file>
    </welcome-file-list>
</web-app>
```

**faces-config.xml** (wystarczy utworzyć w resources/META-INF, będzie wypełnione automatycznie)

### Zadanie z “hello <fname> <sname>”

Create a web application that consists of two pages.
The first one allows a user to provide her or his first and second name.
After submission the user shall be redirected to the second page with a message:
"hello <fname> <sname>", where <fname> and <sname> are first name and second name provided on the first page respectively.
Additional requirements:

- both of the web pages must be developed using JSF technology
- in case a user visits second page before providing data on the first page, the page should display an appropriate message

Struktura

```
MyWebApp/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   └── com/
│   │   │       └── example/
│   │   │           └── mywebapp/
│   │   │               ├── beans/
│   │   │               │   └── UserBean.java
│   │   ├── resources/
│   │   │   └── META-INF/
│   │   │       └── faces-config.xml
│   │   ├── webapp/
│   │   │   ├── index.xhtml
│   │   │   ├── greeting.xhtml
│   │   │   └── WEB-INF/
│   │   │       └── web.xml
└── pom.xml
```

**pom.xml i faces-config.xml** (nie trzeba go modyfikować, domyślnie wygenerowany będzie ok – w razie czego można skorzystać z kodu z poprzednich przykładów)

**UserBean.java**

```
package com.tmq.hello.beans;

import jakarta.enterprise.context.SessionScoped;
import jakarta.inject.Named;

import java.io.Serializable;

@Named
@SessionScoped
public class UserBean implements Serializable {
    private String firstName;
    private String secondName;
    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getSecondName() {
        return secondName;
    }

    public void setSecondName(String secondName) {
        this.secondName = secondName;
    }

    public String submit() {
        if (firstName == null || firstName.isEmpty() || secondName == null || 
secondName.isEmpty()) {
            return "error"; // Redirect to error page if data is missing
        }
        return "greeting?faces-redirect=true"; // Redirect to greeting page
    }

    public boolean isDataProvided() {
        return firstName != null && !firstName.isEmpty() && secondName != null 
&& !secondName.isEmpty();
    }

    public String getGreetingMessage() {
        return "Hello " + firstName + " " + secondName;
    }
}
```

**index.xhtml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
    <title>Enter Your Name</title>
</h:head>
<h:body>
    <h1>Enter Your First and Second Name</h1>
    <h:form>
        <h:outputLabel for="firstName" value="First Name:"/>
        <h:inputText id="firstName" value="#{userBean.firstName}" 
required="true" requiredMessage="First Name is required"/>
        <br/><br/>
        <h:outputLabel for="secondName" value="Second Name:"/>
        <h:inputText id="secondName" value="#{userBean.secondName}" 
required="true" requiredMessage="Second Name is required"/>
        <br/><br/>
        <h:commandButton value="Submit" action="#{userBean.submit}"/>
        <br/><br/>
        <h:message for="firstName" style="color:red;"/>
        <h:message for="secondName" style="color:red;"/>
    </h:form>
</h:body>
</html>
```

**greeting.xhtml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
    <title>Greeting Page</title>
</h:head>
<h:body>
    <h1>Greeting</h1>
    <h:panelGroup rendered="#{userBean.dataProvided}">
        <p>#{userBean.greetingMessage}</p>
    </h:panelGroup>
    <h:panelGroup rendered="#{not userBean.dataProvided}">
        <p>You must provide your first and second name on the previous page.</p>
        <h:link outcome="index" value="Go Back"/>
    </h:panelGroup>
</h:body>
</html>
```

**web.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="6.0" xmlns="https://jakarta.ee/xml/ns/jakartaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd">
    <context-param>
        <param-name>jakarta.faces.PROJECT_STAGE</param-name>
        <param-value>Development</param-value>
    </context-param>
    <servlet>
        <servlet-name>Faces Servlet</servlet-name>
        <servlet-class>jakarta.faces.webapp.FacesServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Faces Servlet</servlet-name>
        <url-pattern>*.xhtml</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>index.xhtml</welcome-file>
    </welcome-file-list>
</web-app>
```

## Zadanie z serwletami i cookies

Create a web application that consists of two web pages.
Upon visiting the first one, a cookie should be stored in user's web browser.
The cookie expires in 30 seconds.
The second web page displays time remaining until the cookie expires.
If a user visits the second page without a cookie stored in a browser, then the second page should display a brief message saying so.
Additional requirements:

- both web pages must be developed using plain servlet technology

**Poniższa aplikacja <u>zazwyczaj</u> działa. Kod jest poprawny, podkreślone *zazwyczaj* zależy od humorów Jakarty EE, konstelacji gwiazd, pogody w Ugandzie i aktualnego losowania Lotto. Zazwyczaj pomaga drugie wciśnięcie przycisku ▶ po starcie serwera.**

Struktura

```
MyWebApp/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   └── com/
│   │   │       └── example/
│   │   │           └── mywebapp/
│   │   │               ├── FirstServlet.java
│   │   │               └── SecondServlet.java
│   │   ├── webapp/
│   │   │   └── WEB-INF/
│   │   │       └── web.xml
└── pom.xml
```

**pom.xml** (trzeba dodać zależność *Jakarta.servlet-api*)

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.tmq</groupId>
    <artifactId>CookieSetCheck</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>war</packaging>
    <name>CookieSetCheck-1.0-SNAPSHOT</name>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <jakartaee>10.0.0</jakartaee>
    </properties>

    <dependencies>
        <dependency>
            <groupId>jakarta.platform</groupId>
            <artifactId>jakarta.jakartaee-api</artifactId>
            <version>${jakartaee}</version>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>jakarta.servlet</groupId>
            <artifactId>jakarta.servlet-api</artifactId>
            <version>6.1.0</version>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.1</version>
                <configuration>
                    <source>11</source>
                    <target>11</target>
                </configuration>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-war-plugin</artifactId>
                <version>3.3.2</version>
            </plugin>
        </plugins>
    </build>
</project>
```

**FirstServlet.java**

```
package com.tmq.cookiesetcheck;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;
import java.time.Instant;

@WebServlet("/first")
public class FirstServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse 
response) throws ServletException, IOException {
        // Create a cookie with the current timestamp
        String cookieName = "visitTime";
        String cookieValue = Instant.now().toString();

        Cookie cookie = new Cookie(cookieName, cookieValue);
        cookie.setMaxAge(30); // Expires in 30 seconds
        cookie.setPath("/"); // Make the cookie accessible across the entire app

        // Add the cookie to the response
        response.addCookie(cookie);

        // Send a simple message to the user
        response.setContentType("text/html");
        response.getWriter().println("<h1>Cookie has been set!</h1>");
        response.getWriter().println("<p>Visit <a href='second'>/second</a> to 
check the remaining time.</p>");
    }
}
```

**SecondServlet.java**

```
package com.tmq.cookiesetcheck;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;
import java.time.Duration;
import java.time.Instant;

@WebServlet("/second")
public class SecondServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        Cookie[] cookies = request.getCookies();
        String visitTime = null;

        if (cookies != null) {
            for (Cookie cookie : cookies) {
                if ("visitTime".equals(cookie.getName())) {
                    visitTime = cookie.getValue();
                    break;
                }
            }
        }

        response.setContentType("text/html");

        if (visitTime == null) {
            // If the cookie is not found, display a message
            response.getWriter().println("<h1>No cookie found!</h1>");
            response.getWriter().println("<p>Please visit <a
href='first'>first</a> to set the cookie.</p>");
        } else {
            Instant cookieInstant = Instant.parse(visitTime);
            Instant now = Instant.now();
            Duration duration = Duration.between(now,
cookieInstant.plusSeconds(30));

            if (duration.isNegative()) {
                // Cookie has already expired
                response.getWriter().println("<h1>Cookie has expired!</h1>");
                response.getWriter().println("<p>Please visit <a
href='first'>first</a> to set a new cookie.</p>");
            } else {
                // Display the remaining time
                long secondsRemaining = duration.getSeconds();
                response.getWriter().println("<h1>Time Remaining:</h1>");
                response.getWriter().println("<p>" + secondsRemaining + "
seconds left until the cookie expires.</p>");
            }
        }
    }
}
```

**index.html** (nie jest obligatoryjny, choć warto dać dla spójności)

```
<!DOCTYPE html>
<html>
    <head>
        <title>Start Page</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <p>Visit <a href="first">first</a> to set a cookie.</p>
    </body>
</html>
```

**web.xml** (bez zmian)

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="https://jakarta.ee/xml/ns/jakartaee"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd"
   version="6.0">
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <display-name>CookieApp</display-name>
</web-app>
```

## Zadanie z losowymi liczbami

Create a web application that consists of two pages.
The first one allows a user to provide an integer number.
After submission, whenever user visits the second page, she or he should see different random number greater than provided on the first page.
Additional requirements:
- both of the web pages must be developed using JSF technology
- provided inputs are session scoped
- in case a user visits second page before providing number on the first page, second page should display a brief message.

Struktura

```
MyWebApp/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   └── com/
│   │   │       └── example/
│   │   │           └── mywebapp/
│   │   │               ├── beans/
│   │   │               │   └── NumberBean.java
│   │   ├── resources/
│   │   │   └── META-INF/
│   │   │       └── faces-config.xml
│   │   ├── webapp/
│   │   │   ├── index.xhtml
│   │   │   ├── random.xhtml
│   │   │   └── WEB-INF/
│   │   │       └── web.xml
└── pom.xml
```

**pom.xml** (tu bez zmian, ewentualnie można dodać zależność *jakarta.faces-api*, choć nie trzeba)

**NumberBean.java**

```
package com.tmq.randomnumbers.beans;

import jakarta.enterprise.context.SessionScoped;
import jakarta.inject.Named;

import java.io.Serializable;
import java.util.Random;

@Named
@SessionScoped
public class NumberBean implements Serializable {
    private Integer userNumber;

    public Integer getUserNumber() {
        return userNumber;
    }

    public void setUserNumber(Integer userNumber) {
        this.userNumber = userNumber;
    }
    public String submit() {
        if (userNumber == null || userNumber <= 0) {
            return "error"; // Redirect to error page if input is invalid
        }
        return "random?faces-redirect=true"; // Redirect to the random page
    }

    public boolean isNumberProvided() {
        return userNumber != null && userNumber > 0;
    }

    public int getRandomNumber() {
        if (!isNumberProvided()) {
            throw new IllegalStateException("User number must be provided before
generating a random number.");
        }
        Random random = new Random();
        return userNumber + 1 + random.nextInt(100); // Generate a random number
greater than userNumber
    }
}
```

**index.xhtml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
    <title>Enter a Number</title>
</h:head>
<h:body>
    <h1>Enter an Integer Number</h1>
    <h:form>
        <h:outputLabel for="userNumber" value="Number:"/>
        <h:inputText id="userNumber" value="#{numberBean.userNumber}"
required="true" requiredMessage="Number is required"/>
        <br/><br/>
        <h:commandButton value="Submit" action="#{numberBean.submit}"/>
        <br/><br/>
        <h:message for="userNumber" style="color:red;"/>
    </h:form>
</h:body>
</html>
```

**random.xhtml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
    <title>Random Number</title>
</h:head>
<h:body>
    <h1>Random Number</h1>
    <h:panelGroup rendered="#{numberBean.numberProvided}">
        <p>Random number greater than #{numberBean.userNumber}:
#{numberBean.randomNumber}</p>
        <h:link outcome="random" value="Generate Another Random Number"/>
    </h:panelGroup>
    <h:panelGroup rendered="#{not numberBean.numberProvided}">
        <p>You must provide a number on the previous page.</p>
        <h:link outcome="index" value="Go Back"/>
    </h:panelGroup>
</h:body>
</html>
```

**web.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="6.0" xmlns="https://jakarta.ee/xml/ns/jakartaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd">
    <context-param>
        <param-name>jakarta.faces.PROJECT_STAGE</param-name>
        <param-value>Development</param-value>
    </context-param>
    <servlet>
        <servlet-name>Faces Servlet</servlet-name>
        <servlet-class>jakarta.faces.webapp.FacesServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Faces Servlet</servlet-name>
        <url-pattern>*.xhtml</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>index.xhtml</welcome-file>
    </welcome-file-list>
</web-app>
```

**Zadanie z parzystością liczb i trzema stronami (treść pisana z pamięci, może się różnić)**

Create a web application that consists of three pages.
The first one allows a user to provide an integer number.
After submission, if the number is even, user should be redirected to the second page containing the word “yolo”, otherwise, if the number is odd, redirection should point to third page with the word “swag”.
Additional requirements:
- all the web pages must be developed using JSF technology
- provided inputs are session scoped
- in case a user visits second or third page before providing number on the first page, those pages should display a brief message.

Struktura

```
MyWebApp/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   └── com/
│   │   │       └── example/
│   │   │           └── mywebapp/
│   │   │               ├── beans/
│   │   │               │   └── NumberBean.java
│   │   ├── resources/
│   │   │   └── META-INF/
│   │   │       └── faces-config.xml
│   │   ├── webapp/
│   │   │   ├── index.xhtml
│   │   │   ├── yolo.xhtml
│   │   │   ├── swag.xhtml
│   │   │   └── WEB-INF/
│   │   │       └── web.xml
└── pom.xml
```

**pom.xml** (bez zmian, można dodać zależność *jakarta.faces-api* jak się komuś nudzi)

**NumberBean.java**

```
package com.tmq.evenorodd.beans;


import jakarta.enterprise.context.SessionScoped;
import jakarta.inject.Named;

import java.io.Serializable;

@Named
@SessionScoped
public class NumberBean implements Serializable {
    private Integer userNumber;

    public Integer getUserNumber() {
        return userNumber;
    }
    public void setUserNumber(Integer userNumber) {
        this.userNumber = userNumber;
    }

    public String submit() {
        if (userNumber == null) {
            return "error"; // Redirect to error page if input is invalid
        }
        return (userNumber % 2 == 0) ? "yolo?faces-redirect=true" : "swag?faces-
redirect=true";
    }

    public boolean isNumberProvided() {
        return userNumber != null;
    }
}
```

**index.xhtml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
    <title>Enter a Number</title>
</h:head>
<h:body>
    <h1>Enter an Integer Number</h1>
    <h:form>
        <h:outputLabel for="userNumber" value="Number:"/>
        <h:inputText id="userNumber" value="#{numberBean.userNumber}"
required="true" requiredMessage="Number is required"/>
        <br/><br/>
        <h:commandButton value="Submit" action="#{numberBean.submit}"/>
        <br/><br/>
        <h:message for="userNumber" style="color:red;"/>
    </h:form>
</h:body>
</html>
```

**yolo.xhtml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
    <title>YOLO</title>
</h:head>
<h:body>
    <h1>YOLO Page</h1>
    <h:panelGroup rendered="#{numberBean.numberProvided and
numberBean.userNumber % 2 == 0}">
        <p>yolo</p>
    </h:panelGroup>
    <h:panelGroup rendered="#{not numberBean.numberProvided or
numberBean.userNumber % 2 != 0}">
        <p>You must provide an even number on the first page.</p>
        <h:link outcome="index" value="Go Back"/>
    </h:panelGroup>
</h:body>
</html>
```

**swag.xhtml**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
<h:head>
    <title>SWAG</title>
</h:head>
<h:body>
    <h1>SWAG Page</h1>
    <h:panelGroup rendered="#{numberBean.numberProvided and 
numberBean.userNumber % 2 != 0}">
        <p>swag</p>
    </h:panelGroup>
    <h:panelGroup rendered="#{not numberBean.numberProvided or 
numberBean.userNumber % 2 == 0}">
        <p>You must provide an odd number on the first page.</p>
        <h:link outcome="index" value="Go Back"/>
    </h:panelGroup>
</h:body>
</html>
```

**web.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="6.0" xmlns="https://jakarta.ee/xml/ns/jakartaee" 
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" 
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee 
https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd">
    <context-param>
        <param-name>jakarta.faces.PROJECT_STAGE</param-name>
        <param-value>Development</param-value>
    </context-param>
    <servlet>
        <servlet-name>Faces Servlet</servlet-name>
        <servlet-class>jakarta.faces.webapp.FacesServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Faces Servlet</servlet-name>
        <url-pattern>*.xhtml</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>index.xhtml</welcome-file>
    </welcome-file-list>
</web-app>
```