



# **Politechnika Śląska**

Wydział Matematyki Stosowanej

## **Języki skryptowe**

Dokumentacja projektu „Drabinki turniejowe”

Aleksandra Starowicz,  
Grzegorz Grochowski,  
Informatyka semestr 3  
Wydział Matematyki Stosowanej

## Spis treści

Opis zadania: .....	3
Opis użytych technologii: .....	3
Użyte biblioteki:.....	3
Opis działania: .....	3
Struktura plików projektu: .....	5
Schemat przepływu danych w projekcie: .....	7
Testy: .....	8
Kody źródłowe:.....	10
DataOperator.java .....	10
TurniejeProjektApplication.java .....	12
Zawodnik.java.....	13
KatModel.java .....	15
HomeController.java .....	15
DrabinkaController.java .....	16
FormularzController.java.....	17
Drabinka.css .....	18
Strona_glowna.css.....	24
Java.css .....	24
Drabinka.html.....	25
Index.html .....	29
Java.html .....	30
Podsumowanie i wnioski: .....	32
Możliwości rozbudowy projektu: .....	32

## Opis zadania:

Zadaniem było stworzenie programu, który wyświetlać będzie na stronie internetowej drabinki turniejowe z uwzględnieniem pozycji zawodnika w rankingu na podstawie zdobytych dotychczas punktów. W naszym projekcie założyliśmy również możliwość rejestracji nowych zawodników do turnieju. Nasz projekt zawiera odpowiedni podział zawodników według: wieku, kategorii wagowej oraz płci.

## Opis użytych technologii:

W celu spełnienia postawionych sobie wymagań co do programu, zdecydowaliśmy się na wykorzystanie Spring-a, popularnego Frameworku dla języka Java pozwalającego na obsługę struktury, zapytań i szablonów stron internetowych. Ma on strukturę modułową, gdzie każdy moduł zajmuje się określonym aspektem aplikacji. Framework pozwala także na łatwą rozbudowę projektów poprzez automatyczne wykrywanie i wdrażanie komponentów. Do obsługi szablonów stron użyliśmy silnika Thymeleaf. Dodatkowo w celu przechowania danych zawodników zdecydowaliśmy się na skorzystanie z zewnętrznego serwera baz danych na silniku MariaDB.

## Użyte biblioteki:

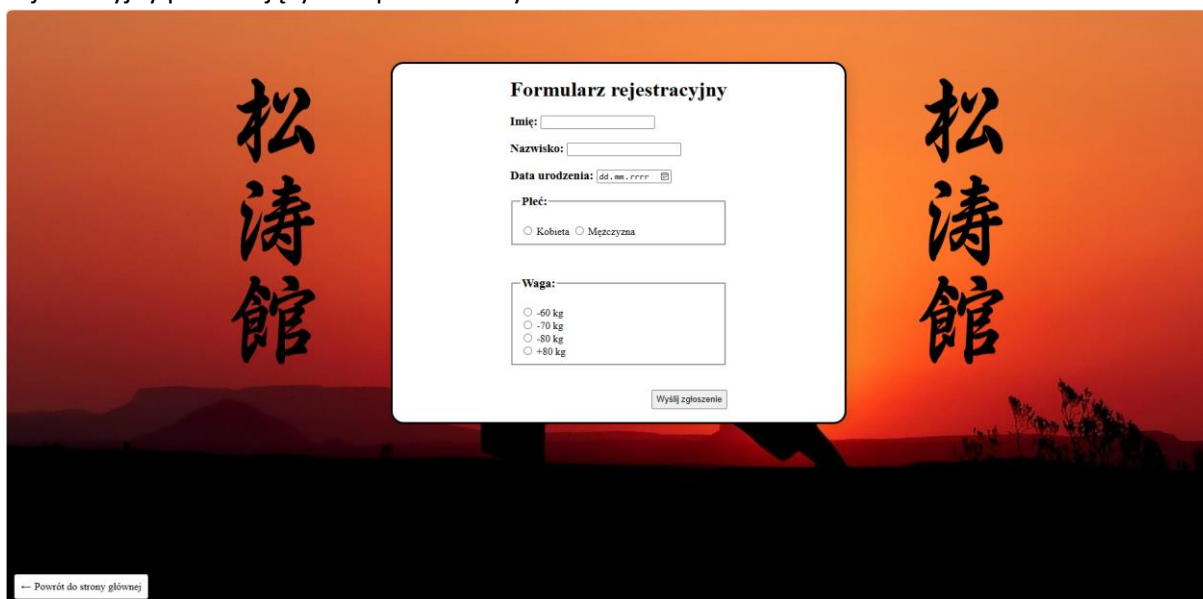
- Jakarta moduł: validation – walidacja danych formularzy
- org.Springframework – obsługa modeli i współdziałania z przeglądarkami
- java.sql – współpraca z bazą danych

## Opis działania:

Do uruchomienia programu niezbędny jest serwer Tomcat oraz uruchomiony serwer baz danych. Po przejściu pod adres strony, jesteśmy witani przez stronę główną, na której możemy wybrać jedną z dwóch opcji: „Zobacz drabinki turniejowe” oraz „Zapisz zawodnika”, przekierowujące na odpowiednie podstrony.



Po wybraniu opcji „Zapisz zawodnika” przekierowani zostajemy na stronę zawierającą formularz rejestracyjny pozwalający na wpisanie danych zawodnika.



The registration form is titled "Formularz rejestracyjny" and is set against a background featuring the Japanese characters "松涛館" (Matsutaikan) on either side of a central white form box. The form contains the following fields:

- Imię: [text input]
- Nazwisko: [text input]
- Data urodzenia: [dd.mm.rrrr] [calendar icon]
- Płeć: [radio buttons for Kobieta and Mężczyzna]
- Waga: [radio buttons for -60 kg, -70 kg, -80 kg, and +80 kg]

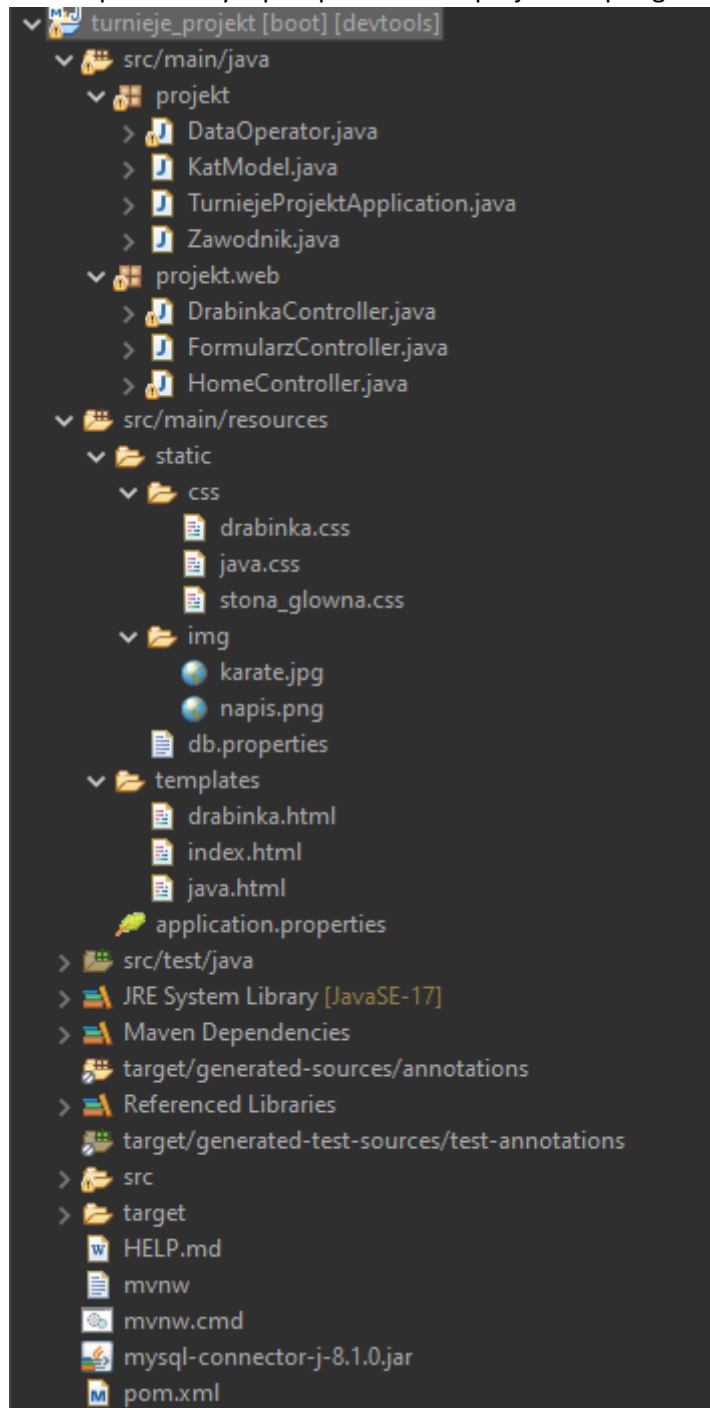
At the bottom right of the form is a button labeled "Wyślij zgłoszenie". At the bottom left of the page is a link: "← Powrót do strony głównej".

Natomiast opcja „Zobacz drabinki turniejowe” przekierowuje na podstronę, która po wybraniu odpowiedniej opcji u dołu strony pozwala na wyświetlenie drabinki turniejowej w wybranej kategorii.



## Struktura plików projektu:

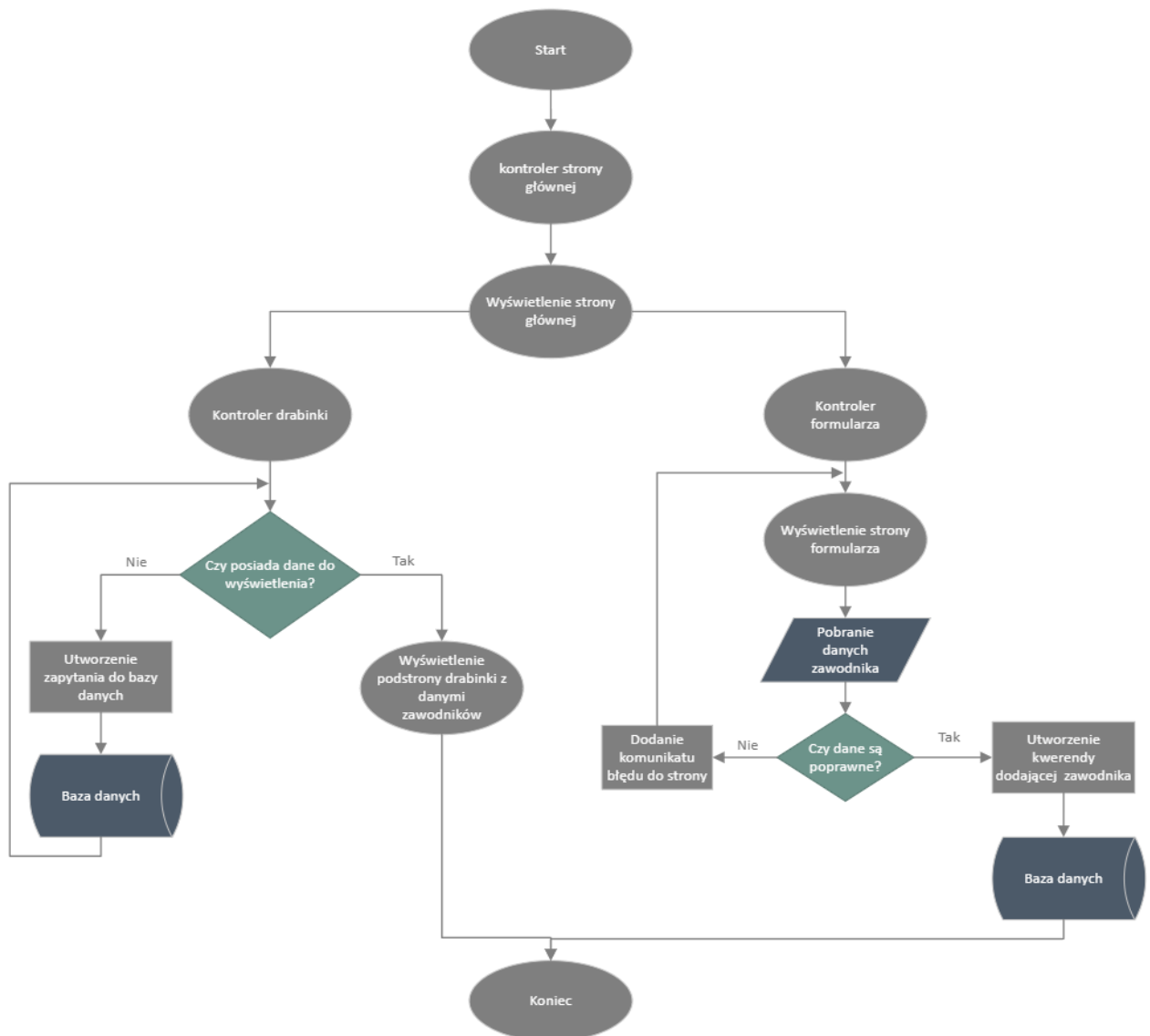
Struktura plików i katalogów samego projektu oparta jest o standardowy szablon projektu Apache Maven poszerzony o pliki potrzebne w projekcie Spring Boot.



- DataOperator.java – Jest to najważniejsza klasa po stronie serwera. Odpowiada ona bezpośrednio za formułowanie zapytań do bazy danych. Z pomocą 3 funkcji zwraca odpowiednią zawartość bazy, sprawdza możliwość dodania nowego zawodnika oraz dodaje z odpowiednimi danymi nowego zawodnika przekazanego przez formularz rejestracyjny. Zawodnicy pobrani z bazy są automatycznie grupowani według kategorii wiekowej i wagowej w strukturze TreeMap.

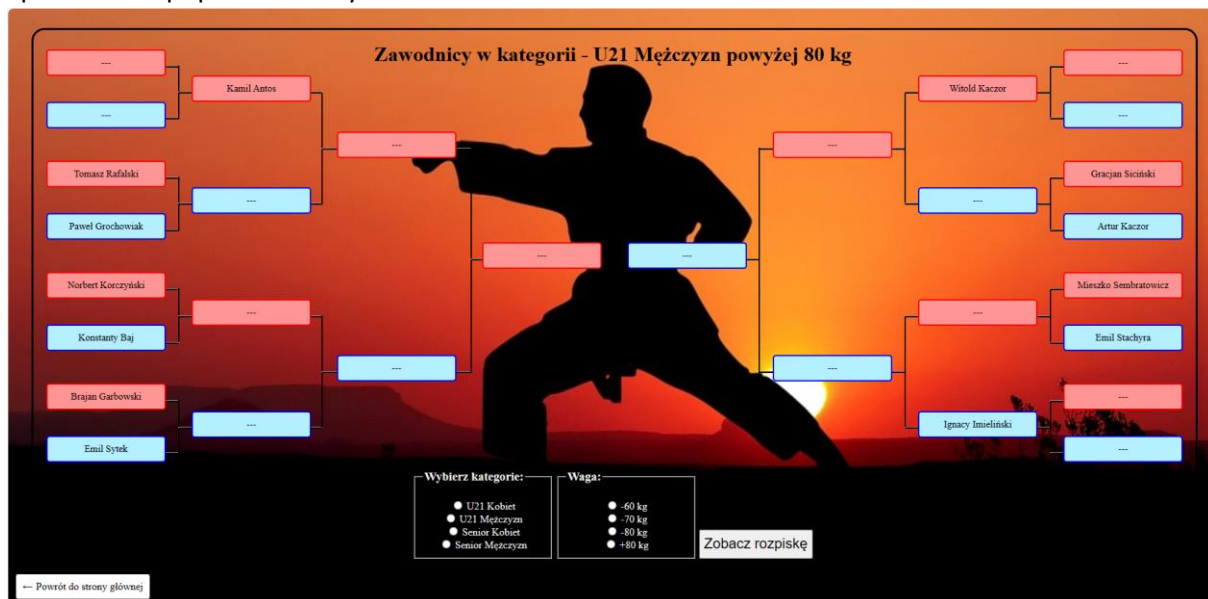
- KatModel.java – jest klasą pomocniczą przechowującą wybraną na podstronie „drabinka” kategorię wagową i wiekową do wyświetlenia.
- TurniejProjektApplication.java – jest to klasa główna projektu, której zadaniem jest inicjalizowanie aplikacji i jej komponentów.
- Zawodnik – Jest to klasa reprezentująca zawodnika ze wszystkimi swoimi danymi. Zawiera dodatkowo adnotacje określające wymagania co do pól zawodnika w celu walidacji poprawności.
- DrabinkaController.java – Jest to klasa kontrolera monitorująca zapytania do podstrony „/drabinka” i wyświetlająca szablon „drabinka.html”. Wykorzystuje obiekt DataOperator w celu pobierania danych zawodników z bazy i dodaje tak pobrane informacje do Modelu strony. Reaguje także na przesłanie formularza wyboru kategorii u dołu strony, zmieniając odpowiednio zawartość strony.
- FormularzController.java – Monitoruje zapytania do podstrony „/formularz” i wyświetla szablon „java.html”. Przeprowadza walidację wprowadzonych do formularza danych i w zależności od wyniku zwraca błędy wyświetlane pod odpowiednimi polami formularza lub jeżeli wszystko przebiegło poprawnie dodaje zawodnika do bazy za pośrednictwem obiektu DataOperator.
- HomeController.java – Monitoruje zapytania do strony głównej i wyświetla stronę „index.html”
- Drabinka.html - zawiera kod strony do wyświetlania drabinki oraz odpowiada za odpowiednie umieszczenie zawodników, którzy są dostarczani w postaci posortowanej listy. Domyślnie wyświetla ich w skrajnie lewej i prawej kolumnie, jeśli jednak ilość zawodników jest mniejsza i nie pozwala na wypełnienie wszystkich pól przesuwa zawodników do środka w kierunku finału. Zawodnicy są rozmieszczeni w taki sposób, aby zawodnicy z największą ilością punktów rankingowych mieli jak najmniej walk oraz walczyli na początku ze "słabszymi" zawodnikami.
- Index.html - zawiera kod strony głównej
- Java.html - zawiera kod strony wyświetlającej formularz rejestracyjny, który obsługuje ewentualne błędy jego wypełnienia. Błędy są wyświetlane w postaci komunikatów.
- Drabinka.css - zawiera style potrzebne do strony drabinka.html
- Java.css - zawiera style potrzebne do strony java.html
- Strona\_glowna.css - zawiera style potrzebna do strony index.html
- db.properties – dane dostępne do bazy danych

## Schemat przepływu danych w projekcie:



## Testy:

- Sprawdzenie poprawności wyświetlania drabinki:



## Baza danych:

<div><div><div><div></div><div></div><div></div></div><div></div></div></div>														ID	Imie	Nazwisko	DataUr	Plec	Waga	Punkty	1	Kategoria
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	72	Emil	Sytek	1996-01-11	M	87	1	U21											
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	79	Konstanty	Baj	2005-08-01	M	81	2	U21											
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	87	Artur	Kaczor	2007-02-09	M	90	3	U21											
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	63	Emil	Stachyra	1996-10-31	M	98	6	U21											
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	26	Paweł	Grochowiak	2006-10-29	M	91	6	U21											
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	31	Tomasz	Rafalski	1996-10-18	M	98	8	U21											
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	9	Mieszko	Sembratowicz	2001-06-28	M	86	14	U21											
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	28	Gracjan	Siciński	2005-02-28	M	98	17	U21											
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	92	Norbert	Korczyński	2008-01-01	M	96	22	U21											
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	75	Brajan	Garbowski	2007-05-12	M	85	24	U21											
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	55	Witold	Kaczor	1990-06-13	M	87	25	U21											
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	82	Ignacy	Imieliński	1994-06-12	M	82	26	U21											
<div><div><div></div><div></div><div></div></div><div></div></div>	<div><div></div><div></div><div></div></div> <div>Edytuj</div>	<div><div></div><div></div><div></div></div> <div>Kopiuj</div>	<div><div></div><div></div><div></div></div> <div>Usuń</div>	45	Kamil	Antos	2006-03-31	M	90	29	U21											



- Rejestracja nowego zawodnika:

**Formularz rejestracyjny**

Imię:

Nazwisko:

Data urodzenia:

Płeć:

☒ Kobieta ☐ Mężczyzna

Waga:

☐ -60 kg  
☒ -70 kg  
☐ -80 kg  
☐ +80 kg

[Wyślij zgłoszenie](#)

[← Powrót do strony głównej](#)

Applikacja discord.com udostępnia Twój ekran. [Zatrzymaj udostępnianie](#) [Ukryj](#)

baza przed rejestracją:

	ID	Imie	Nazwisko	DataUr	Plec	Waga	Punkty	Kategoria
<input type="checkbox"/> <a href="#">Edytuj</a> <a href="#">Kopiuj</a> <a href="#">Usuń</a>	12	Katarzyna	Strzoda	2008-01-01	K	62	9	SNR
<input type="checkbox"/> <a href="#">Edytuj</a> <a href="#">Kopiuj</a> <a href="#">Usuń</a>	18	Vanessa	Gombrowicz	2001-10-25	K	70	8	SNR

baza po rejestracji:

	ID	Imie	Nazwisko	DataUr	Plec	Waga	Punkty	Kategoria
<input type="checkbox"/> <a href="#">Edytuj</a> <a href="#">Kopiuj</a> <a href="#">Usuń</a>	12	Katarzyna	Strzoda	2008-01-01	K	62	9	SNR
<input type="checkbox"/> <a href="#">Edytuj</a> <a href="#">Kopiuj</a> <a href="#">Usuń</a>	18	Vanessa	Gombrowicz	2001-10-25	K	70	8	SNR
<input type="checkbox"/> <a href="#">Edytuj</a> <a href="#">Kopiuj</a> <a href="#">Usuń</a>	1019	Adrianna	Nowak	1996-07-12	K	70	0	SNR

- Puste pola formularza:

**Formularz rejestracyjny**

Imię:

Imię nie może być puste

Nazwisko:

Nazwisko nie może być puste

Data urodzenia:

Proszę podać datę urodzenia

Płeć:

☐ Kobieta ☐ Mężczyzna

Płeć jest obowiązkowa

Waga:

☐ -60 kg  
☐ -70 kg  
☐ -80 kg  
☐ +80 kg

Proszę zaznaczyć kategorię wagową

[Wyślij zgłoszenie](#)

[← Powrót do strony głównej](#)

Applikacja discord.com udostępnia Twój ekran. [Zatrzymaj udostępnianie](#) [Ukryj](#)

- Przepełniona kategoria zawodników  
16 zawodniczek w danej kategorii:

←T→			▼	ID	Imie	Nazwisko	DataUr	Plec	Waga	Punkty	Kategoria
<input type="checkbox"/>				12	Katarzyna	Strzoda	2008-01-01	K	62	9	SNR
<input type="checkbox"/>				18	Vanessa	Gombrowicz	2001-10-25	K	70	8	SNR
<input type="checkbox"/>				1019	Adrianna	Nowak	1996-07-12	K	70	0	SNR
<input type="checkbox"/>				1020	Ala	Kowalska	1998-04-12	K	70	0	SNR
<input type="checkbox"/>				1021	Daria	Dąb	1998-02-12	K	70	0	SNR
<input type="checkbox"/>				1022	Klaudia	Prosta	1996-06-23	K	70	0	SNR
<input type="checkbox"/>				1023	Ela	Nowak	1992-09-23	K	70	0	SNR
<input type="checkbox"/>				1024	Iza	Nowakowska	1991-05-30	K	70	0	SNR
<input type="checkbox"/>				1025	Ola	Ząb	1996-10-30	K	70	0	SNR
<input type="checkbox"/>				1026	Natalia	Soja	1994-12-24	K	70	0	SNR
<input type="checkbox"/>				1027	Marta	Michnik	1991-06-27	K	70	0	SNR
<input type="checkbox"/>				1028	Michalina	Ząbek	2000-02-12	K	70	0	SNR
<input type="checkbox"/>				1029	Anastazja	Zięba	2001-11-11	K	70	0	SNR
<input type="checkbox"/>				1030	Anna	Nowacka	2001-02-12	K	70	0	SNR
<input type="checkbox"/>				1031	Alina	Krysiak	1996-02-12	K	70	0	SNR
<input type="checkbox"/>				1032	Natasza	Nowa	1994-12-12	K	70	0	SNR

Komunikat:

## Kody źródłowe:

### DataOperator.java

```
package projekt;

import java.io.IOException;
import java.io.InputStream;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.sql.*;
import java.util.*;

public class DataOperator
{
    private TreeMap<Zawodnik.Kategoria, ArrayList<Zawodnik>> mapa = new TreeMap<>();
```

```

//wartości domyślne
private String driver = "com.mysql.cj.jdbc.Driver";
private String url = "";
private String user = "root";
private String password = "";

public boolean insertData(Zawodnik zaw)
{
    int result=1;
    try(Connection con = getConnection(); Statement stat = con.createStatement())
    {
        String kat = zaw.getKategoria().toString().substring(0,3);
        result = stat.executeUpdate(String.format("INSERT INTO zawodnicy VALUES
(%s,'%s','%s','%s','%s',%d,%d,'%s');",
        "null",zaw.getImie(),zaw.getNazwisko(),zaw.getUrodzenie().toString(),zaw.getPlec(),zaw.getWaga(
),zaw.getPunkty(),kat));
    }
    catch (SQLException | IOException e)
    {
        System.out.println(e.getMessage());
        e.printStackTrace();
        return false;
    }

    return (result > 0);
}

public boolean canAdd(Zawodnik zaw)
{
    try(Connection con = getConnection(); Statement stat = con.createStatement())
    {
        int waga_D = 0;
        int waga_G = 0;

        if(zaw.getWaga()<=60) waga_G=60;
        else if(zaw.getWaga()<=70) {waga_D = 60; waga_G = 70;}
        else if(zaw.getWaga()<=80) {waga_D = 70; waga_G = 80;}
        else if(zaw.getWaga()>80 || zaw.getWaga()!=-1) {waga_D = 80; waga_G = 1000;}

        ResultSet rs = stat.executeQuery("SELECT Count(ID) FROM zawodnicy WHERE
kategoria='"+zaw.getKategoria().toString().substring(0, 3)+"' and
plec='"+zaw.getPlec()+"' and waga BETWEEN "
+waga_D+" AND "+waga_G+";");
        rs.next();
        return rs.getInt(1)<16;
    }
    catch(SQLException e)
    {
        System.out.println(e.getMessage());
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }

    return false;
}

public TreeMap<Zawodnik.Kategoria, ArrayList<Zawodnik>> getData()
{
    for(Zawodnik.Kategoria k : Zawodnik.Kategoria.values())
        mapa.put(k, new ArrayList<>());

    try(Connection con = getConnection(); Statement stat = con.createStatement())
    {
        String[] s = {"U21", "SNR"};
        String kat = "";
        for(String k : s)
        {
            ResultSet rs = stat.executeQuery("SELECT * FROM zawodnicy WHERE
Kategoria='"+k.toString()+"'");

```

```

        while(rs.next())
        {
            int waga = rs.getInt("waga");
            if(waga<=60) {waga = 60;}else if(waga<=70) {waga=70;} else
            kat = rs.getString("Kategoria")+"_"+((waga==
if(waga <= 80) {waga = 80;}else {waga = -1;};
1)?"W":waga)+rs.getString("Plec");
            Zawodnik z = new
            Zawodnik(rs.getString("Imie"),rs.getString("Nazwisko"),rs.getString("DataUr"),rs.getString("Plec"),rs.g
            etInt("Waga"),rs.getInt("Punkty"),Zawodnik.Kategoria.valueOf(kat));
            mapa.get(Zawodnik.Kategoria.valueOf(kat)).add(z);
        }
    }

    for(Zawodnik.Kategoria k : Zawodnik.Kategoria.values())
    {
        Collections.sort(mapa.get(k),Collections.reverseOrder());
    }
}
catch(SQLException e)
{
    System.out.println(e.getMessage());
}
catch(Exception e)
{
    e.printStackTrace();
}

return mapa;
}

private void korekta(ArrayList<Zawodnik> lista)
{
    int rozmiar = lista.size();
    for(int i=rozmiar;i<16;i++)
        lista.add(new Zawodnik());
}

public Connection getConnection() throws SQLException, IOException
{
    Properties props = new Properties();
    try(InputStream in =
Files.newInputStream(Paths.get("src/main/resources/static/db.properties")))
    {
        props.load(in);
    }

    if(props.containsKey("jdbc.user"))
    {
        driver = props.getProperty("jdbc.driver");
        url = props.getProperty("jdbc.engine") + "://" +
props.getProperty("jdbc.host")+":"+props.getProperty("jdbc.port")+"/"+props.getProperty("jdbc.dbname");
        user = props.getProperty("jdbc.user");
        password = props.getProperty("jdbc.password");
    }
    return DriverManager.getConnection(url,user,password);
}
}

```

## TurniejeProjektApplication.java

```

package projekt;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class TurniejeProjektApplication {

    public static void main(String[] args) {
        SpringApplication.run(TurniejeProjektApplication.class, args);
    }

}

```

## Zawodnik.java

```
package projekt;

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;

import jakarta.validation.constraints.Min;
import jakarta.validation.constraints.NotBlank;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Past;

public class Zawodnik implements Comparable<Object>
{
    public enum Kategoria{U21_60K, U21_70K, U21_80K, U21_WK, SNR_60K, SNR_70K, SNR_80K, SNR_WK,
U21_60M, U21_70M, U21_80M, U21_WM, SNR_60M, SNR_70M, SNR_80M, SNR_WM};

    @NotBlank(message="Imie nie może być puste")
    private String imie;
    @NotBlank(message="Nazwisko nie może być puste")
    private String nazwisko;
    @NotNull(message="Proszę podać datę urodzenia")
    @Past(message="Data nie może być z przyszłości")
    private LocalDate urodzenie;
    @NotBlank(message="Płeć jest obowiązkowa")
    private String plec;
    @Min(value = 0, message="Proszę zaznaczyć kategorię wagową")
    private int waga;
    private int punkty;
    private Kategoria kat;

    public Zawodnik()
    {
        imie = "";
        nazwisko = "";
        waga = 0;
        punkty = 0;
        plec = "";
        urodzenie = null;
        kat = null;
    }

    public Zawodnik(String imie, String nazwisko, String data, String plec, int waga, int punkty,
Kategoria kat)
    {
        this.imie = imie;
        this.nazwisko = nazwisko;
        this.urodzenie = LocalDate.parse(data, DateTimeFormatter.ISO_LOCAL_DATE);
        this.plec = plec;
        this.waga = waga;
        this.punkty = punkty;
        this.kat = kat;
    }

    public Zawodnik(String imie, String nazwisko, String data, String plec, int waga, int punkty,
String kat)
    {
        this(imie,nazwisko,data,plec,waga,punkty,Kategoria.valueOf(kat));
    }

    public String toString()
    {
        return imie+" "+nazwisko+" "+urodzenie+" pkt: "+punkty;
    }

    public int compareTo(Object o)
    {
        if(this == o) return 0;
        if(o == null) return -1;
        if(!(o instanceof Zawodnik)) return 1;
        Zawodnik zw = (Zawodnik) o;
    }
}
```

```

        return this.punkty-zw.punkty;
    }

    public String getImie()
    {
        return imie;
    }

    public String getNazwisko()
    {
        return nazwisko;
    }

    public String getPlec()
    {
        return plec;
    }

    public LocalDate getUrodzenie()
    {
        return urodzenie;
    }

    public int getWiek()
    {
        return (int) LocalDate.now().until(urodzenie, ChronoUnit.YEARS);
    }

    public int getWaga()
    {
        return waga;
    }

    public int getPunkty()
    {
        return punkty;
    }

    public Kategoria getKategoria()
    {
        return kat;
    }

    public void setImie(String imie)
    {
        this.imie = imie;
    }

    public void setNazwisko(String nazwisko)
    {
        this.nazwisko = nazwisko;
    }

    public void setUrodzenie(LocalDate ld)
    {
        this.urodzenie = ld;
    }

    public void setUrodzenie(String ld)
    {
        this.urodzenie = LocalDate.parse(ld, DateTimeFormatter.ISO_LOCAL_DATE);
    }

    public void setPlec(String plec)
    {
        this.plec = plec;
    }

    public void setWaga(int waga)
    {
        this.waga = waga;
    }

    public void setPunkty(int punkty)

```

```

    {
        this.punkty = punkty;
    }

    public void setKategoria(Kategoria kat)
    {
        this.kat = kat;
    }
}

```

## KatModel.java

```

package projekt;

public class KatModel
{
    private String kat;
    private String waga;

    public KatModel(String kat, String waga)
    {
        this.kat = kat;
        this.waga = waga;
    }

    public KatModel() {}

    public void setKat(String kat)
    {
        this.kat = kat;
    }

    public void setWaga(String waga)
    {
        this.waga = waga;
    }

    public String getKat()
    {
        return kat;
    }

    public String getWaga()
    {
        return waga;
    }
}

```

## HomeController.java

```

package projekt.web;

import java.util.ArrayList;
import java.util.TreeMap;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import projekt.*;

@Controller
public class HomeController
{
    @GetMapping("/")
    public String home()
    {
        return "index.html";
    }
}

```

## DrabinkaController.java

```
package projekt.web;

import java.util.ArrayList;
import java.util.TreeMap;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.SessionAttributes;

import jakarta.validation.Valid;
import projekt.*;

@Controller
@SessionAttributes("katmodel")
public class DrabinkaController
{
    private static Zawodnik.Kategoria kat = Zawodnik.Kategoria.U21_60K;
    private String kat_w = "U21 ";
    private String plec_w = "Kobiety ";
    private String waga_w = "do 60 kg";
    DataOperator op;

    @GetMapping("/drabinka")
    public String drabinka(Model model)
    {
        op = new DataOperator();
        TreeMap<Zawodnik.Kategoria, ArrayList<Zawodnik>> mapa = op.getData();
        model.addAttribute("wyswietlana", kat_w+plec_w+waga_w);
        model.addAttribute("zawodnicy", mapa.get(kat));
        model.addAttribute("katmodel", new KatModel());
        return "drabinka";
    }

    @PostMapping("/drabinka")
    public String create(KatModel katmodel, Model model)
    {
        if(katmodel.getKat()==null)
            katmodel = new KatModel("U21_k", "60");
        op = new DataOperator();
        StringBuilder sb = new StringBuilder();
        sb.append(katmodel.getKat().substring(0, 4));
        sb.append(katmodel.getWaga());
        sb.append(katmodel.getKat().substring(4).toUpperCase());
        kat = Zawodnik.Kategoria.valueOf(sb.toString());
        TreeMap<Zawodnik.Kategoria, ArrayList<Zawodnik>> mapa = op.getData();

        kat_w = katmodel.getKat().substring(0, 3);
        plec_w = (katmodel.getKat().substring(4, 5).equals("k"))?" Kobiet ":" Mężczyzn ";
        waga_w = (katmodel.getWaga().equals("W"))?"powyżej 80 kg":"do "+katmodel.getWaga()+"
kg";

        model.addAttribute("zawodnicy", mapa.get(kat));
        model.addAttribute("wyswietlana", kat_w+plec_w+waga_w);
        return "drabinka";
    }
}
```



## FormularzController.java

```
package projekt.web;

import java.time.LocalDate;
import java.time.Period;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.Errors;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.SessionAttributes;

import jakarta.validation.Valid;
import projekt.*;

@Controller
@SessionAttributes("zawodnik")
public class FormularzController
{
    @GetMapping("/formularz")
    public String form(Model model)
    {
        model.addAttribute("zawodnik", new Zawodnik());
        model.addAttribute("przepelnienie", false);
        return "java";
    }

    @PostMapping("/formularz")
    public String dodaj(@Valid Zawodnik zawodnik, Errors errors, Model model)
    {
        if(errors.hasErrors())
            return "java";

        StringBuilder sb = new StringBuilder();
        DataOperator op = new DataOperator();
        int lata = Period.between(LocalDate.now(), zawodnik.getUrodzenie()).getYears();
        if(lata>21)
            sb.append("SNR_");
        else
            sb.append("U21_");

        if(zawodnik.getWaga()==1)
            sb.append("W");
        else
            sb.append(zawodnik.getWaga());
        sb.append(zawodnik.getPlec());
        zawodnik.setKategoria(Zawodnik.Kategoria.valueOf(sb.toString()));

        if(op.canAdd(zawodnik))
        {
            op.insertData(zawodnik);
            return "redirect:/";
        }
        else
            model.addAttribute("przepelnienie", true);
        return "java";
    }
}
```

## Drabinka.css

```
body {
  background-image: url("../img/karate.jpg");
  background-size: cover;
}
#buttons{
  margin-left:auto;
  margin-right:auto;
  width:35%;
}
fieldset{
  margin-top: 0px;
  display:inline-block;
  width: 10%;
}
legend{
  color:white;
}
.nagLowek{
  z-index: 3;
  position:absolute;
  text-align: center;
  left: 50%;
  transform: translateX(-50%);
  margin-top: -10px;
}
.radiopanel{
  float:left;
  padding: 0 3%;
}
label{
  color:white;
}
.wrapper {
  display: flex;
  justify-content: center;
  width: 1800px;
  margin: 30px 28px -30px 28px;
  box-shadow: 6px 6px 10px 0px rgba(0, 0, 0, 0.1);
  border-radius: 20px;
  border-top: 3px solid black;
  border-left:3px solid black ;
  border-right: 3px solid black ;
  z-index: 2;
  position: relative;
  font-size: 15px;
}
.ramki {
  z-index: 1;
  position: absolute;
  width: 1600px;
  margin: 20px 140px;
}
#nagLowek {
  margin: 8px;
  width: 936px;
  height: 200px;
}
.kolumna {
  float: Left;
  margin: 20px;
  width: 2000px;
  height: 650px;
}
.ramka1 {
  background-color: #ff9696;
  padding: 10px;
  border-width: 2px;
  border-style: solid;
  border-color: red;
}
```

```

border-radius: 5px;
margin-bottom: 40px;
margin-top: 10px;
text-align: center;
}
.ramka2 {
background-color: #b4f0ff;
padding: 10px;
border-width: 2px;
border-style: solid;
border-color: blue;
border-radius: 5px;
margin-bottom: 50px;
text-align: center;
}
.ramka3 {
background-color: #ff9696;
padding: 10px;
border-width: 2px;
border-style: solid;
border-color: red;
border-radius: 5px;
margin-bottom: 132px;
margin-top: 50px;
text-align: center;
}
.ramka4 {
background-color: #b4f0ff;
padding: 10px;
border-width: 2px;
border-style: solid;
border-color: blue;
border-radius: 5px;
margin-bottom: 132px;
text-align: center;
}
.ramka5 {
background-color: #ff9696;
padding: 10px;
border-width: 2px;
border-style: solid;
border-color: red;
border-radius: 5px;
margin-bottom: 304px;
margin-top: 137px;
text-align: center;
}
.ramka6 {
background-color: #b4f0ff;
padding: 10px;
border-width: 2px;
border-style: solid;
border-color: blue;
border-radius: 5px;
text-align: center;
}
.ramka7 {
background-color: #ff9696;
padding: 10px;
border-width: 2px;
border-style: solid;
border-color: red;
border-radius: 5px;
margin-top: 308px;
text-align: center;
}
.ramka8 {
background-color: #b4f0ff;
padding: 10px;
border-width: 2px;
border-style: solid;
border-color: blue;
border-radius: 5px;
margin-top: 308px;

```

```

    text-align: center;
}
.button2 {
    position: absolute;
    bottom: 0;
    left: 0;
    margin: 10px;
    padding: 10px;
    border: 2px solid black;
    border-radius: 5px;
    background-color: white;
    color: black;
    text-decoration: none;
}
.button2::before {
    content: '\2190 '; /* Znak unicode strzałki w lewo */
    margin-right: 5px;
}
.button {
    margin-top: 30px;
    margin-bottom: 30px;
    height: 45px;
    font-size: 22px;
}

.wybor1 {
    margin-right: 40px;
}

.white-text {
    color: white;
    font-size: 20px;
}

.linia1,
.linia2,
.linia3,
.linia4,
.linia5,
.linia6,
.linia7,
.linia8 {
    border-bottom: 2px solid black;
    width: 25px;
    left: 90px;
    position: absolute;
}

.linia1,
.linia1_12 {
    margin-top: 38px;
}

.linia2,
.linia2_12 {
    margin-top: 118px;
}

.linia3,
.linia3_12 {
    margin-top: 210px;
}

.linia4,
.linia4_12 {
    margin-top: 291px;
}

.linia5,
.linia5_12 {
    margin-top: 382px;
}

.linia6,
.linia6_12 {
    margin-top: 465px;
}

.linia7,

```

```

.linia7_12 {
    margin-top: 555px;
}
.linia8,
.linia8_12 {
    margin-top: 635px;
}

.pionowa1,
.pionowa2,
.pionowa3,
.pionowa4 {
    border-left: 2px solid black;
    height: 83px;
    width: 0;
    left: 115px;
    position: absolute;
}
.pionowa1,
.pionowa1_6 {
    margin-top: 38px;
}

.pionowa2,
.pionowa2_6 {
    margin-top: 210px;
}

.pionowa3,
.pionowa3_6 {
    margin-top: 382px;
}

.pionowa4,
.pionowa4_6 {
    margin-top: 555px;
}

.linia1_3,
.linia2_3,
.linia3_3,
.linia4_3 {
    border-bottom: 2px solid black;
    width: 25px;
    left: 312px;
    position: absolute;
}
.linia1_2,
.linia2_2,
.linia3_2,
.linia4_2 {
    border-bottom: 2px solid black;
    width: 25px;
    left: 115px;
    position: absolute;
}

.linia1_2,
.linia1_3,
.linia1_10,
.linia1_11 {
    margin-top: 78px;
}

.linia2_2,
.linia2_3,
.linia2_10,
.linia2_11 {
    margin-top: 250px;
}
.linia3_2,
.linia3_3,
.linia3_10,
.linia3_11 {

```

```

    margin-top: 424px;
}
.linia4_2,
.linia4_3,
.linia4_10,
.linia4_11 {
    margin-top: 596px;
}

.pionowa1_2,
.pionowa2_2 {
    border-left: 2px solid black;
    height: 173px;
    width: 0;
    left: 337px;
    position: absolute;
}
.pionowa1_2,
.pionowa1_5 {
    margin-top: 80px;
}
.pionowa2_2,
.pionowa2_5 {
    margin-top: 424px;
}

.linia1_4,
.linia2_4 {
    border-bottom: 2px solid black;
    width: 25px;
    left: 337px;
    position: absolute;
}
.linia1_4,
.linia1_5,
.linia1_8,
.linia1_9 {
    margin-top: 165px;
}
.linia2_4,
.linia2_5,
.linia2_8,
.linia2_9 {
    margin-top: 510px;
}
.linia1_5,
.linia2_5 {
    border-bottom: 2px solid black;
    width: 25px;
    left: 543px;
    position: absolute;
}

.pionowa1_3 {
    border-left: 2px solid black;
    height: 348px;
    width: 0;
    left: 568px;
    position: absolute;
}

.pionowa1_3,
.pionowa1_4 {
    margin-top: 165px;
}

.linia1_6 {
    border-bottom: 2px solid black;
    width: 25px;
    left: 568px;
    position: absolute;
}

.linia1_6,

```

```

.linia1_7 {
  margin-top: 336px;
}

.linia1_7 {
  border-bottom: 2px solid black;
  width: 25px;
  left: 988px;
  position: absolute;
}

.linia1_8,
.linia2_8 {
  border-bottom: 2px solid black;
  width: 25px;
  left: 1014px;
  position: absolute;
}

.linia1_9,
.linia2_9 {
  border-bottom: 2px solid black;
  width: 25px;
  left: 1214px;
  position: absolute;
}

.pionowa1_4 {
  border-left: 2px solid black;
  height: 348px;
  width: 0;
  left: 1014px;
  position: absolute;
}

.pionowa1_5,
.pionowa2_5 {
  border-left: 2px solid black;
  height: 173px;
  width: 0;
  left: 1240px;
  position: absolute;
}

.linia1_10,
.linia2_10,
.linia3_10,
.linia4_10 {
  border-bottom: 2px solid black;
  width: 25px;
  left: 1240px;
  position: absolute;
}

.linia1_11,
.linia2_11,
.linia3_11,
.linia4_11 {
  border-bottom: 2px solid black;
  width: 25px;
  left: 1440px;
  position: absolute;
}

.pionowa1_6,
.pionowa2_6,
.pionowa3_6,
.pionowa4_6 {
  border-left: 2px solid black;
  height: 83px;
  width: 0;
  left: 1465px;
  position: absolute;
}

.linia1_12,
.linia2_12,

```

```

.linia3_12,
.linia4_12,
.linia5_12,
.linia6_12,
.linia7_12,
.linia8_12 {
  border-bottom: 2px solid black;
  width: 25px;
  left: 1465px;
  position: absolute;
}

```

## Strona\_glowna.css

```

body {
  background-image: url("../img/karate.jpg");
  background-size: cover;
  text-align: center;
}

.napis1 {
  margin-right: 50px;
}
.napis2 {
  margin-left: 50px;
}

.container {
  display: flex;
  justify-content: space-evenly;
  width: 1600px;
  margin: 80px 140px;
}
.buttons{
  margin-top:100px;
  display: flex;
  flex-direction: column;
  align-items: center;
}

.button{
  text-decoration: none;
  margin-bottom:20px;
  padding: 10px 20px;
  font-size: 16px;
  text-align: center;
  background-color: lightgray;
  color: black;
  border-radius: 5px;
  cursor: pointer;
}

.button:hover {
  background-color: grey;
}

```

## Java.css

```

h1 {
  text-align: center;
  margin-bottom: 30px;
}

body {
  background-image: url("../img/karate.jpg");
  background-size: cover;
}
.error{
  color:red;
  margin-top:-30px;
  font-size: 12px;
  font-weight: normal
}
fieldset{

```



```

        margin-top:-20px;
    }

    .button2 {
        position: absolute;
        bottom: 0;
        left: 0;
        margin: 10px;
        padding: 10px;
        border: 2px solid black;
        border-radius: 5px;
        background-color: white;
        color: black;
        text-decoration: none;
    }
    .button2::before {
        content: '\2190 '; /* Znak unicode strzałki w lewo */
        margin-right: 5px;
    }
    .wrapper {
        display: flex;
        justify-content: center;
        width: 700px;
        margin: 0 20px;
        box-shadow: 6px 6px 20px 0px rgba(0, 0, 0, 0.1);
        background: white;
        border-radius: 20px;
        border-radius: 20px;
        border: 3px solid black;
    }

    .napis1 {
        margin-right: 50px;
        max-height:500px;
    }
    .napis2 {
        margin-left: 50px;
        max-height:500px;
    }

    .container {
        display: flex;
        justify-content: center;
        width: 1600px;
        margin: 80px 140px;
    }

    .button {
        margin-top: 20px;
        margin-bottom: 20px;
        height: 30px;
        text-align: center;
        display: block;
        margin-left: auto;
    }

```

## Drabinka.html

```

<!DOCTYPE html>
<html
    lang="pl"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:th="http://www.thymeleaf.org"
>
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Zawodnicy w kategoriach</title>
        <link rel="preconnect" href="https://fonts.googleapis.com" />
        <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
        <link
            href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap"
            rel="stylesheet"
        />

```

```

<link th:href="@{css/drabinka.css}" rel="stylesheet" />
</head>

<body>
<div class="naglowek">
<h1><p>Zawodnicy w kategorii - <span th:text="{wyswietlana}"></span></p></h1>
</div>

<div class="ramki">
<hr class="linia1" />
<hr class="linia2" />
<hr class="linia3" />
<hr class="linia4" />
<hr class="linia5" />
<hr class="linia6" />
<hr class="linia7" />
<hr class="linia8" />
<div class="pionowa1"></div>
<div class="pionowa2"></div>
<div class="pionowa3"></div>
<div class="pionowa4"></div>
<hr class="linia1_2" />
<hr class="linia2_2" />
<hr class="linia3_2" />
<hr class="linia4_2" />
<hr class="linia1_3" />
<hr class="linia2_3" />
<hr class="linia3_3" />
<hr class="linia4_3" />
<div class="pionowa1_2"></div>
<div class="pionowa2_2"></div>
<hr class="linia1_4" />
<hr class="linia2_4" />
<hr class="linia1_5" />
<hr class="linia2_5" />
<div class="pionowa1_3"></div>
<hr class="linia1_6" />
<hr class="linia1_7" />
<hr class="linia1_8" />
<hr class="linia2_8" />
<div class="pionowa1_4"></div>
<hr class="linia1_9" />
<hr class="linia2_9" />
<div class="pionowa1_5"></div>
<div class="pionowa2_5"></div>
<hr class="linia1_10" />
<hr class="linia2_10" />
<hr class="linia3_10" />
<hr class="linia4_10" />
<hr class="linia1_11" />
<hr class="linia2_11" />
<hr class="linia3_11" />
<hr class="linia4_11" />
<div class="pionowa1_6"></div>
<div class="pionowa2_6"></div>
<div class="pionowa3_6"></div>
<div class="pionowa4_6"></div>
<hr class="linia1_12" />
<hr class="linia2_12" />
<hr class="linia3_12" />
<hr class="linia4_12" />
<hr class="linia5_12" />
<hr class="linia6_12" />
<hr class="linia7_12" />
<hr class="linia8_12" />
</div>

<div class="wrapper">
<div class="kolumna">
<div class="ramka1">
<center th:if="{zawodnicy.size()}>15}" th:text="{zawodnicy[0].imie + ' ' +
zawodnicy[0].nazwisko}">---</center>
<center th:if="{zawodnicy.size()}<=15}">---</center>
</div>

```

```

<div class="ramka2">
  <center th:if="{zawodnicy.size()}>15}" th:text="{zawodnicy[15].imie + ' ' +
zawodnicy[15].nazwisko}">---</center>
  <center th:if="{zawodnicy.size()}<=15}">---</center>
</div>
<div class="ramka1">
  <center th:if="{zawodnicy.size()}>8}" th:text="{zawodnicy[7].imie + ' ' +
zawodnicy[7].nazwisko}">---</center>
  <center th:if="{zawodnicy.size()}<=8}">---</center>
</div>
<div class="ramka2">
  <center th:if="{zawodnicy.size()}>8}" th:text="{zawodnicy[8].imie + ' ' +
zawodnicy[8].nazwisko}">---</center>
  <center th:if="{zawodnicy.size()}<=8}">---</center>
</div>
<div class="ramka1">
  <center th:if="{zawodnicy.size()}>11}" th:text="{zawodnicy[4].imie + ' ' +
zawodnicy[4].nazwisko}">---</center>
  <center th:if="{zawodnicy.size()}<=11}">---</center>
</div>
<div class="ramka2">
  <center th:if="{zawodnicy.size()}>11}" th:text="{zawodnicy[11].imie + ' ' +
zawodnicy[11].nazwisko}">---</center>
  <center th:if="{zawodnicy.size()}<=11}">---</center>
</div>
<div class="ramka1">
  <center th:if="{zawodnicy.size()}>12}" th:text="{zawodnicy[3].imie + ' ' +
zawodnicy[3].nazwisko}">---</center>
  <center th:if="{zawodnicy.size()}<=12}">---</center>
</div>
<div class="ramka2">
  <center th:if="{zawodnicy.size()}>12}" th:text="{zawodnicy[12].imie + ' ' +
zawodnicy[12].nazwisko}">---</center>
  <center th:if="{zawodnicy.size()}<=12}">---</center>
</div>
</div>
<div class="kolumna">
  <div class="ramka3">
    <center th:if="{zawodnicy.size()}>7 and zawodnicy.size() < 16}" th:text="{zawodnicy[0].imie
+ ' ' + zawodnicy[0].nazwisko}">---</center>
    <center th:if="{zawodnicy.size()}<=7 or zawodnicy.size() >= 16}">---</center>
  </div>
  <div class="ramka4">
    <center th:if="{zawodnicy.size()}>7 and zawodnicy.size() < 9}" th:text="{zawodnicy[7].imie +
' ' + zawodnicy[7].nazwisko}">---</center>
    <center th:if="{zawodnicy.size()}<=7 or zawodnicy.size() >= 9}">---</center>
  </div>
  <div class="ramka3">
    <center th:if="{zawodnicy.size()}>4 and zawodnicy.size() < 12}" th:text="{zawodnicy[4].imie
+ ' ' + zawodnicy[4].nazwisko}">---</center>
    <center th:if="{zawodnicy.size()}<=4 or zawodnicy.size() >= 12}">---</center>
  </div>
  <div class="ramka4">
    <center th:if="{zawodnicy.size()}>4 and zawodnicy.size() < 13}" th:text="{zawodnicy[3].imie
+ ' ' + zawodnicy[3].nazwisko}">---</center>
    <center th:if="{zawodnicy.size()}<=4 or zawodnicy.size() >= 13}">---</center>
  </div>
</div>
<div class="kolumna">
  <div class="ramka5">
    <center th:if="{zawodnicy.size()}>3 and zawodnicy.size() < 8}" th:text="{zawodnicy[0].imie +
' ' + zawodnicy[0].nazwisko}">---</center>
    <center th:if="{zawodnicy.size()}<=3 or zawodnicy.size() >= 8}">---</center>
  </div>
  <div class="ramka6">
    <center th:if="{zawodnicy.size()}>3 and zawodnicy.size() < 5}" th:text="{zawodnicy[3].imie +
' ' + zawodnicy[3].nazwisko}">---</center>
    <center th:if="{zawodnicy.size()}<=3 or zawodnicy.size() >= 5}">---</center>
  </div>
</div>
<div class="kolumna">
  <div class="ramka7">
    <center th:if="{zawodnicy.size()}>0 and zawodnicy.size() <= 3}" th:text="{zawodnicy[0].imie
+ ' ' + zawodnicy[0].nazwisko}">---</center>
  </div>
</div>

```

```

        <center th:if="{zawodnicy.size()}<=0 or zawodnicy.size() > 3}">---</center>
    </div>
</div>
<div class="kolumna">
    <div class="ramka8">
        <center th:if="{zawodnicy.size()}>1 and zawodnicy.size() <=2}" th:text="{zawodnicy[1].imie +
' ' + zawodnicy[1].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=1 or zawodnicy.size() >2}">---</center>
    </div>
</div>
<div class="kolumna">
    <div class="ramka5">
        <center th:if="{zawodnicy.size()}>2 and zawodnicy.size() < 6}" th:text="{zawodnicy[2].imie +
' ' + zawodnicy[2].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=2 or zawodnicy.size() >= 6}">---</center>
    </div>
    <div class="ramka6">
        <center th:if="{zawodnicy.size()}>2 and zawodnicy.size() < 7}" th:text="{zawodnicy[1].imie +
' ' + zawodnicy[1].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=2 or zawodnicy.size() >= 7}">---</center>
    </div>
</div>
<div class="kolumna">
    <div class="ramka3">
        <center th:if="{zawodnicy.size()}>5 and zawodnicy.size() < 14}" th:text="{zawodnicy[2].imie
+ ' ' + zawodnicy[2].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=5 or zawodnicy.size() >= 14}">---</center>
    </div>
    <div class="ramka4">
        <center th:if="{zawodnicy.size()}>5 and zawodnicy.size() < 11}" th:text="{zawodnicy[5].imie
+ ' ' + zawodnicy[5].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=5 or zawodnicy.size() >= 11}">---</center>
    </div>
    <div class="ramka3">
        <center th:if="{zawodnicy.size()}>6 and zawodnicy.size() < 10}" th:text="{zawodnicy[6].imie
+ ' ' + zawodnicy[6].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=6 or zawodnicy.size() >= 10}">---</center>
    </div>
    <div class="ramka4">
        <center th:if="{zawodnicy.size()}>6 and zawodnicy.size() < 15}" th:text="{zawodnicy[1].imie
+ ' ' + zawodnicy[1].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=6 or zawodnicy.size() >= 15}">---</center>
    </div>
</div>
<div class="kolumna">
    <div class="ramka1">
        <center th:if="{zawodnicy.size()}>13}" th:text="{zawodnicy[2].imie + ' ' +
zawodnicy[2].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=13}">---</center>
    </div>
    <div class="ramka2">
        <center th:if="{zawodnicy.size()}>13}" th:text="{zawodnicy[13].imie + ' ' +
zawodnicy[13].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=13}">---</center>
    </div>
    <div class="ramka1">
        <center th:if="{zawodnicy.size()}>10}" th:text="{zawodnicy[5].imie + ' ' +
zawodnicy[5].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=10}">---</center>
    </div>
    <div class="ramka2">
        <center th:if="{zawodnicy.size()}>10}" th:text="{zawodnicy[10].imie + ' ' +
zawodnicy[10].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=10}">---</center>
    </div>
    <div class="ramka1">
        <center th:if="{zawodnicy.size()}>9}" th:text="{zawodnicy[6].imie + ' ' +
zawodnicy[6].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=9}">---</center>
    </div>
    <div class="ramka2">
        <center th:if="{zawodnicy.size()}>9}" th:text="{zawodnicy[9].imie + ' ' +
zawodnicy[9].nazwisko}">---</center>
        <center th:if="{zawodnicy.size()}<=9}">---</center>
    </div>

```

```

        </div>
        <div class="ramka1">
            <center th:if="{zawodnicy.size()}>14}" th:text="{zawodnicy[1].imie + ' ' +
zawodnicy[1].nazwisko}">---</center>
            <center th:if="{zawodnicy.size()}<=14}">---</center>
        </div>
        <div class="ramka2">
            <center th:if="{zawodnicy.size()}>14}" th:text="{zawodnicy[14].imie + ' ' +
zawodnicy[14].nazwisko}">---</center>
            <center th:if="{zawodnicy.size()}<=14}">---</center>
        </div>
    </div>
</div>

<div>
    <div>
        <a href="http://localhost:8080" class="button2">Powrót do strony głównej</a>
    </div>
    <center>
        <form th:object="{katmodel}" method="POST">
            <fieldset>
                <legend><h3>Wybierz kategorie:</h3></legend>
                <div>
                    <input type="radio" th:field="*{kat}" id="U21_k" name="kat" th:value="U21_k"/>
                    <label for="U21_k">U21 Kobiet</label>
                </div>
                <div>
                    <input type="radio" th:field="*{kat}" id="U21_m" name="kat" th:value="U21_m"/>
                    <label for="U21_m">U21 Mężczyzn</label>
                </div>
                <div>
                    <input type="radio" th:field="*{kat}" id="SNR_k" name="kat" th:value="SNR_k"/>
                    <label for="SNR_k">Senior Kobiet</label>
                </div>
                <div>
                    <input type="radio" th:field="*{kat}" id="SNR_m" name="kat" th:value="SNR_m"/>
                    <label for="SNR_m">Senior Mężczyzn</label>
                </div>
            </fieldset>
            <fieldset>
                <legend><h3>Waga:</h3></legend>
                <div>
                    <input type="radio" th:field="*{waga}" id="-60kg" name="waga" th:value="60" />
                    <label for="-60kg">-60 kg </label>
                </div>
                <div>
                    <input type="radio" th:field="*{waga}" id="-70kg" name="waga" th:value="70"/>
                    <label for="-70kg">-70 kg </label>
                </div>
                <div>
                    <input type="radio" th:field="*{waga}" id="-80kg" name="waga" th:value="80"/>
                    <label for="-80kg">-80 kg </label>
                </div>
                <div>
                    <input type="radio" th:field="*{waga}" id="+80kg" name="waga" th:value="W"/>
                    <label for="+80kg">+80 kg </label>
                </div>
            </fieldset>
            <input class="button" type="submit" value="Zobacz rozpisę" />
        </div>
    </form>
</center>
</div>
</body>
</html>

```

## Index.html

```

<!DOCTYPE HTML>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org%22%3E/
<head>
    <meta charset="utf-8"/>

```

```

<title>Drabinki truniejowe</title>
<link th:href="@{css/stona_glowna.css}" rel="stylesheet" />
</head>
<body>
<h1>Witaj na Stronie Głównej</h1>
<div class="container">

<div class="buttons">
<a href="http://localhost:8080/drabinka" class="button">Zobacz drabinki turniejowe</a>
<a href="http://localhost:8080/formularz" class="button">Zapisz zawodnika</a>
</div>

</div>
</div></div>
</body>
</html>

```

## Java.html

```

<!DOCTYPE html>
<html lang="pl" xmlns="http://www.w3.org/1999/xhtml"
xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Rejestracja</title>
<link rel="preconnect" href="https://fonts.googleapis.com" />
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
<link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap"
rel="stylesheet"
/>
<link th:href="@{css/java.css}" rel="stylesheet" />
</head>
<body>
<div class="container">


<div class="wrapper">
<main>
<h1>Formularz rejestracyjny</h1>
<h2 class="error" th:if="{przepelnienie}">Brak miejsc w tej kategorii</h2>
<form
th:object="{zawodnik}"
method="post"
>
<div>
<h3>
Imię: <input type="text" th:field="{imie}" name="Imie" /><br/>
<span class="error" th:if="{#fields.hasErrors('imie')}" th:errors="{imie}">Proszę
podać imię</span>
</h3>
<h3>
Nazwisko:
<input type="text" th:field="{nazwisko}" name="Nazwisko" /><br/>
<span class="error" th:if="{#fields.hasErrors('nazwisko')}"
th:errors="{nazwisko}">Proszę podać nazwisko nazwiska</span>
</h3>
<h3>
Data urodzenia: <input type="date" th:field="{urodzenie}" /><br/>
<span class="error" th:if="{#fields.hasErrors('urodzenie')}"
th:errors="{urodzenie}">Proszę podać date urodzenia</span>
</h3>
<fieldset>
<legend><h3>Płeć:</h3></legend>
<div>
<input
type="radio"
id="kobieta"
name="plec"
value="kobieta"
th:field="{plec}"

```

```

        th:value="'K'"
    />
    <label for="kobieta">Kobieta</label>
    <input
        type="radio"
        id="menczyzna"
        name="plec"
        value="menczyzna"
        th:field="*{plec}"
        th:value="'M'"
    />
    <label for="menczyzna">Mężczyzna</label>
</div>
</fieldset>
    <span class="error" th:if="${#fields.hasErrors('plec')}"
th:errors="*{plec}">Proszę wybrać płeć</span><br/>

<pre id="log"></pre>
<fieldset>
    <legend><h3>Waga:</h3></legend>
    <div>
        <input
            type="radio"
            id="-60kg"
            name="waga"
            value="-60kg"
            th:field="*{waga}"
            th:value="60"
        />
        <label for="-60kg">-60 kg </label>
    </div>
    <div>
        <input
            type="radio"
            id="-70kg"
            name="waga"
            value="-70kg"
            th:field="*{waga}"
            th:value="70"
        />
        <label for="-70kg">-70 kg </label>
    </div>
    <div>
        <input
            type="radio"
            id="-80kg"
            name="waga"
            value="-80kg"
            th:field="*{waga}"
            th:value="80"
        />
        <label for="-80kg">-80 kg </label>
    </div>
    <div>
        <input
            type="radio"
            id="+80kg"
            name="waga"
            value="1"
            th:field="*{waga}"
            th:value="1"
        />
        <label for="+80kg">+80 kg </label>
    </div>
</fieldset>
    <span class="error" th:if="${#fields.hasErrors('waga')}"
th:errors="*{waga}">Proszę wybrać kategorie wagową </span><br/>
    </div>
    <input class="button" type="submit" value="Wyślij zgłoszenie" />
</form>
</main>
</div>


```

```
</div>
<div>
    <a href="http://localhost:8080" class="button2">Powrót do strony głównej</a>
</div>
</body>
</html>
```

## Podsumowanie i wnioski:

Początkowe założenia projektu zostały spełnione. Współpraca z bazą danych, walidacja formularzy, zabezpieczenia przed niepożądanymi działaniami działają poprawnie. Trudności przysporzyło nam prawidłowe połączenie różnych komponentów projektu w jedną całość, odpowiednie formatowanie danych do postaci przyjmowanej przez bazę danych oraz odpowiednia prezentacja danych na samej stronie internetowej.

## Możliwości rozbudowy projektu:

W dalszej perspektywie projekt można rozbudować o moduł do śledzenia poszczególnych walk w czasie rzeczywistym, która automatycznie uzupełniać będzie drabinki.