

Statystyka w informatyce sprawozdanie z realizacji projektu

Maciej Bandura, Marcin Ślusarczyk; 1ID21B

6 listopada 2024

1. Wstęp

Wykonywany temat: Analiza wyników egzaminów studentów z różnych przedmiotów.

W ramach przeprowadzono analizę danych dotyczących ocen uzyskanych przez studentów z wybranych przedmiotów. Celem projektu było opracowanie narzędzia umożliwiającego szczegółową analizę wyników egzaminów oraz ich wizualizację, co pozwoli na lepsze zrozumienie rozkładów ocen oraz zależności między poszczególnymi przedmiotami. W projekcie skupiono się na identyfikacji rodzaju przetwarzanych danych oraz doborze odpowiedniej formy przechowywania, aby zapewnić efektywny dostęp do danych. Następnie, zaprojektowano aplikację, która umożliwi obliczanie podstawowych miar statystycznych, takich jak miary tendencji centralnej i zróżnicowania, oraz wyznaczanie współczynników korelacji pomiędzy parami atrybutów. Dodatkowo, dane zostały przedstawione graficznie przy użyciu różnorodnych wykresów, co ułatwia interpretację wyników i wspiera wyciąganie wniosków.

Wykorzystane technologie: Projekt został oparty na bibliotece React, co umożliwiło stworzenie dynamicznej i interaktywnej aplikacji webowej. W celu przechowywania danych, poszukiwaliśmy rozwiązania typu lite SQL, które pozwoliłoby na obsługę danych lokalnie, bez potrzeby implementowania części backendowej. Wybór padł na Dexie.js, lekką bazę danych SQL-like, dedykowaną dla aplikacji React. Dzięki temu rozwiązaniu możliwe było przechowywanie i szybkie przetwarzanie danych na urządzeniu użytkownika.

Rodzaj przetwarzanych danych: Dane użyte w projekcie nie pochodzą z żadnego zewnętrznego źródła, lecz zostały wygenerowane za pomocą naszego autorskiego generatora ocen. W systemie znajduje się 30 studentów, z których każdy posiada 6 ocen z 6 przedmiotów:

- matematyki,
- informatyki,
- fizyki,
- języka obcego,
- historii,
- wychowania fizycznego,

Generator ocen został zaprojektowany w taki sposób, aby umożliwić uzyskanie realistycznych danych, które mogą posłużyć do przeprowadzenia analizy statystycznej i wizualizacji wyników.

Skala ocen: W projekcie zastosowano skalę ocen stosowaną na studiach wyższych, która obejmuje następujące stopnie:

Ocena	Opis
5.0	Bardzo dobry
4.5	Dobry plus
4.0	Dobry
3.5	Dostateczny plus
3.0	Dostateczny
2.0	Niedostateczny

Tabela 1: Skala ocen stosowana w projekcie

W ramach projektu wyznaczane są następujące wskaźniki statystyczne:

- **Średnia** – miara tendencji centralnej, wskazująca przeciętną wartość ocen.
- **Mediana** – środkowa wartość zestawu danych, dzieląca je na dwie równe części.
- **Dominanta** – najczęściej występująca wartość w zbiorze ocen.
- **Wariancja** – miara rozproszenia danych, wskazująca, jak bardzo oceny odbiegają od średniej.
- **Odchylenie standardowe** – pierwiastek z wariancji, będący bardziej intuicyjną miarą rozrzutu danych.
- **Korelacja Pearsona** – miara zależności liniowej pomiędzy parami ocen z różnych przedmiotów.

Te wskaźniki pozwalają na szczegółową analizę wyników studentów i umożliwiają wyciąganie wniosków na temat rozkładu ocen oraz zależności między przedmiotami.

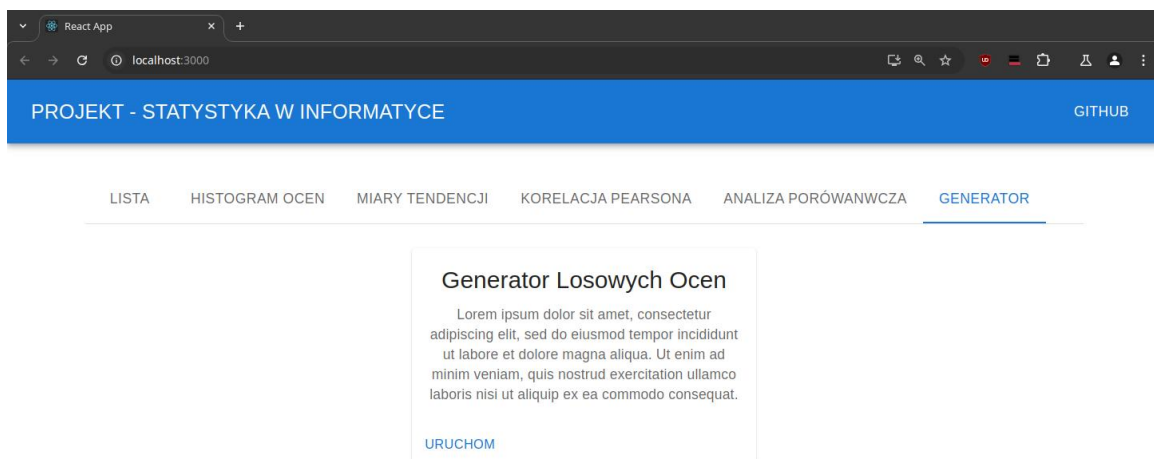
2. Aplikacja

Aplikacja została zrealizowana jako Single-Page Application (SPA), co zapewnia płynne i szybkie działanie bez konieczności przeładowywania stron. Cały interfejs użytkownika stworzono w React, co pozwoliło na efektywne zarządzanie stanem aplikacji i dynamiczne renderowanie komponentów. Dodatkowo wykorzystano bibliotekę MUI (Material-UI), która umożliwiła zastosowanie gotowych, responsywnych komponentów o nowoczesnym wyglądzie, takich jak przyciski, tabele, wykresy i inne elementy interfejsu.

Aplikacja jest zorganizowana w formie kart (tabs), gdzie każda karta odpowiada za inną funkcjonalność, np. wgląd w dane, wyświetlanie miar statystycznych. Dzięki temu układowi użytkownik może intuicyjnie nawigować między różnymi częściami aplikacji. Cała struktura aplikacji została zaprojektowana z myślą o wygodzie użytkownika, z naciskiem na przejrzystość oraz intuicyjną nawigację.

2.1 Pierwsze uruchomienie aplikacji

Po uruchomieniu aplikacji należy przejść do karty **GENERATOR** gdzie, użytkownik ma możliwość wygenerowania losowych ocen dla losowych studentów. Uwaga, zbieżność imion i nazwisk jest przypadkowa.



Rysunek 1: Karta generatora

2.2 Generator ocen

Generator ocen w projekcie został zaprojektowany z myślą o zapewnieniu różnorodności wyników, co miało na celu odwzorowanie realistycznych scenariuszy oceniania studentów. Proces generowania ocen polega na tworzeniu losowych wyników dla 30 studentów, z uwzględnieniem specyficznych reguł dotyczących każdego przedmiotu, co wpływa na dystrybucję ocen w taki sposób, by były one zróżnicowane, ale jednocześnie sensowne.

Generator działa w następujący sposób:

1. **Wstępne przygotowanie:** Na początku każdorazowo czyszczona jest baza danych, aby móc wygenerować nową pulę ocen.
2. **Generowanie ocen:** Dla każdego studenta (30 studentów w projekcie) przypisywane są oceny z 6 przedmiotów. Oceny generowane są losowo, z uwzględnieniem reguł zależnych od przedmiotu:
 - **Informatyka:** Jeśli wylosowana ocena jest wyższa niż 4.0, ale spełniony jest warunek prawdopodobieństwa, ocena jest zmniejszana o 2 punkty.
 - **Język obcy:** Na tym przedmiocie wprowadzone są reguły, które preferują ocenę 5.0 lub 4.5 z większym prawdopodobieństwem, a dla niższych ocen ocena jest ustalana w sposób losowy.
 - **Fizyka:** Wartość oceny zależy od oceny z matematyki – może być taka sama, lub różnić się o niewielką wartość, jeżeli matematyka jest w średnim zakresie ocen.
 - **Matematyka:** Ocena z matematyki jest zapisywana, a potem wykorzystywana do wpływania na inne oceny (np. z fizyki).
 - **Wychowanie fizyczne (WF):** Oceny są poprawiane z minimalnej wartości (2.0) na 3.0, aby zapobiec zbyt niskim wynikom.
3. **Zapis do bazy:** Po wygenerowaniu oceny dla każdego przedmiotu, dane są dodawane do bazy danych, przechowując informacje o studencie, przedmiocie, ocenie i płci studenta.

W JavaScriptie generowanie liczb losowych odbywa się przy użyciu funkcji **Math.random()**. Ta funkcja zwraca pseudolosową liczbę zmiennoprzecinkową w przedziale od 0 (włącznie) do 1 (wyłącznie). Wykorzystując tę funkcję, można łatwo generować liczby w dowolnym przedziale, przekształcając wynik **Math.random()** do pożądanego skali.

```
1  async function generateMarks ()
2  {
3      db.oceny.clear()
4
5      for (let i = 0; i < 30; i++) {
6          const [isMan, student] = getRandomName();
```

```

7   let matma = 2.0;
8
9   for (const przedmiot of _przedmioty) {
10      let losowaOcena = getRandomOcena();
11
12
13      if (przedmiot == "Informatyka" && losowaOcena >= 4.0 && Math.random() < 0.8)
14         ↪ {
15            losowaOcena -= 2.0;
16         }
17
18      if (przedmiot == "Język obcy" && Math.random() < 0.3) {
19         losowaOcena = Math.random() < 0.5 ? 5.0 : 4.5;
20      }
21
22      if (przedmiot == "Język obcy" && losowaOcena <= 3.0 && Math.random() < 0.5)
23         ↪ {
24            losowaOcena = 4.0;
25         }
26
27      if (przedmiot == "Fizyka") {
28         if (Math.random() < 0.5) {
29            losowaOcena = matma;
30         } else {
31            if (matma > 3.0 && matma < 5.0) {
32               losowaOcena = matma + (Math.random() < 0.5 ? -0.5 : 0.5);
33            }
34         }
35      }
36
37      if (losowaOcena == 2.5) {
38         if (Math.random() < 0.5) {
39            losowaOcena -= 0.5;
40         } else {
41            losowaOcena += 0.5;
42         }
43      }
44
45      if (przedmiot == "Matematyka") {
46         matma = losowaOcena;
47      }
48
49      if (przedmiot == "WF" && losowaOcena == 2.0) {
50         losowaOcena = 3.0;
51      }

```

```

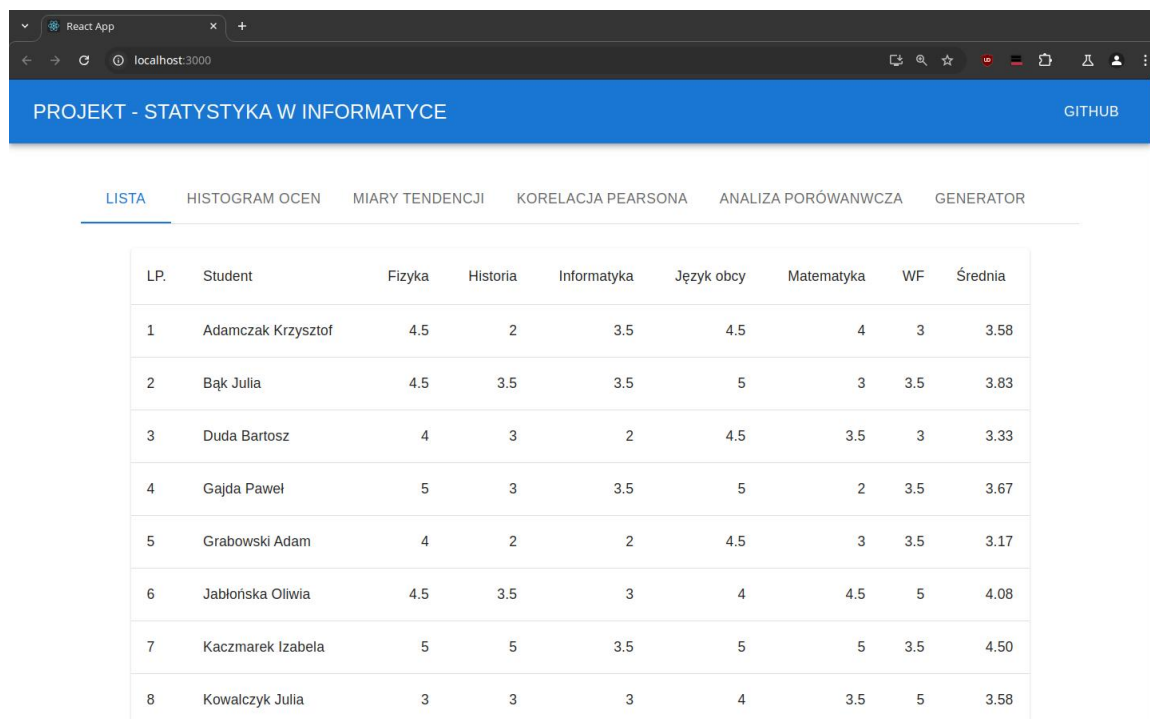
50
51     const ocena = {
52       student: `${student}`,
53       przedmiot: przedmiot,
54       ocena: losowaOcena,
55       plec: isMan
56     }
57
58     await db.oceny.add(ocena);
59   }
60 }
61 }
62

```

Listing 1: Kod generatora ocen

2.3 Lista studentów

Wyniki generatora przedstawiamy w tradycyjny sposób - w tabeli, gdzie umieszczone są oceny każdego studenta z poszczególnych przedmiotów.



LP.	Student	Fizyka	Historia	Informatyka	Język obcy	Matematyka	WF	Średnia
1	Adamczak Krzysztof	4.5	2	3.5	4.5	4	3	3.58
2	Bąk Julia	4.5	3.5	3.5	5	3	3.5	3.83
3	Duda Bartosz	4	3	2	4.5	3.5	3	3.33
4	Gajda Paweł	5	3	3.5	5	2	3.5	3.67
5	Grabowski Adam	4	2	2	4.5	3	3.5	3.17
6	Jabłońska Oliwia	4.5	3.5	3	4	4.5	5	4.08
7	Kaczmarek Izabela	5	5	3.5	5	5	3.5	4.50
8	Kowalczyk Julia	3	3	3	4	3.5	5	3.58

Rysunek 2: Lista studentów wraz z ich ocenami

```

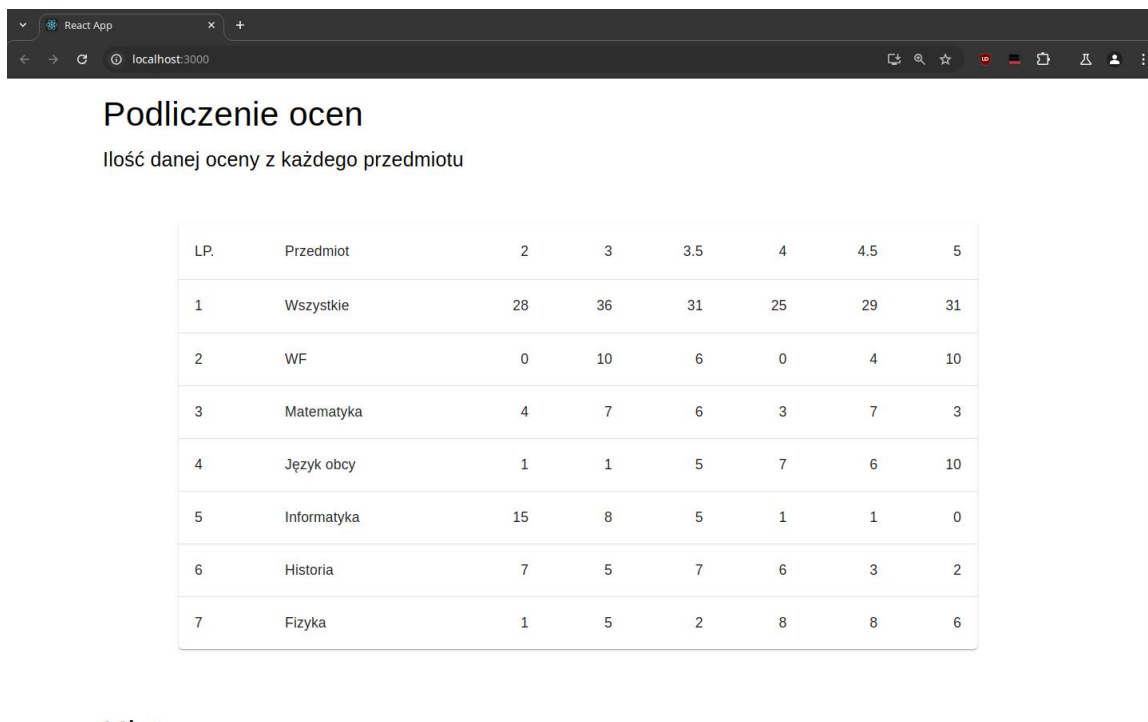
1   async function fetchOceny()
2   {
3       await db.oceny; // preload
4       const allPrzedmioty = await
      ↪ db.oceny.orderBy('przedmiot').uniqueKeys(); // .sort();
5
6       const sortedPrzedmioty = allPrzedmioty.sort();
7       setPrzedmioty(sortedPrzedmioty);
8
9       const dane = [];
10      const studenci = await db.oceny.orderBy('student').uniqueKeys();
11      const sortedStudenci = studenci.sort();
12
13      /** Pobieramy oceny studenta */
14      for (const student of sortedStudenci) {
15          const ocenyForStudent = await db.oceny
16              .where("student")
17              .equals(student)
18              .sortBy("przedmiot");
19
20          dane.push({
21              student: student,
22              oceny: ocenyForStudent.map(n => n.ocena)
23          });
24      }
25
26      console.log(dane);
27      setOceny(dane);
28  }
29

```

Listing 2: Utworzenie obiektu do renderowania w tabelce

2.4 Histogram ocen

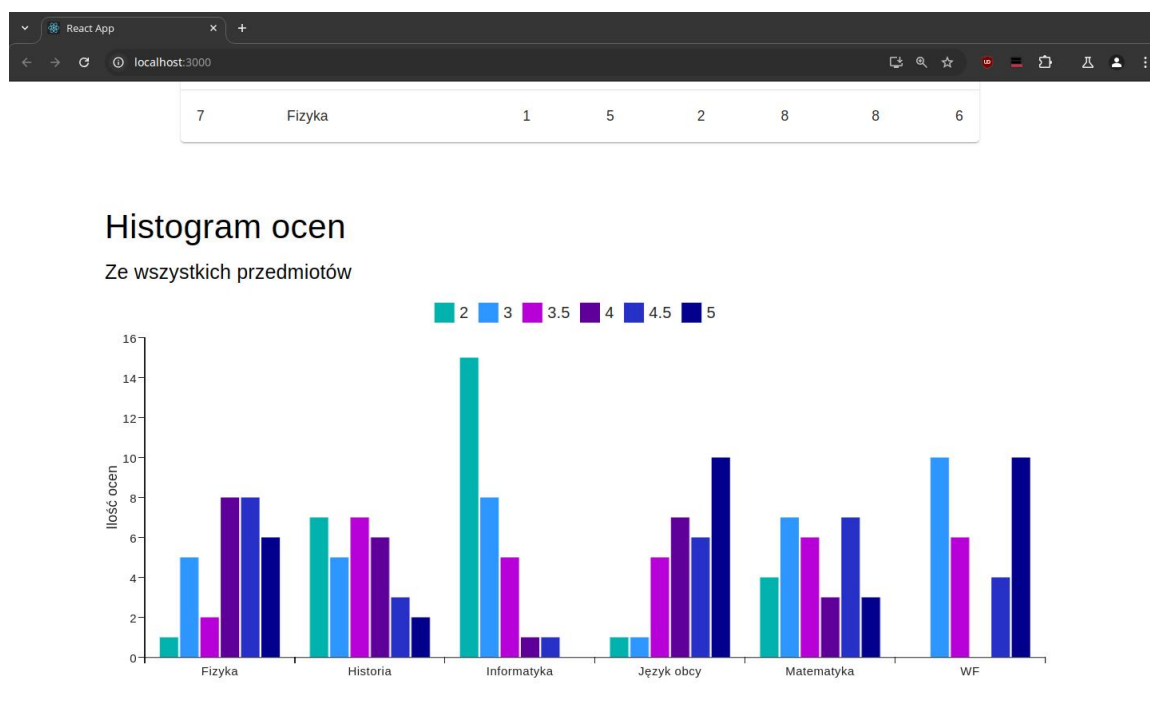
Histogram ocen jest jednym z podstawowych narzędzi wizualizacyjnych w analizie danych, umożliwiającym przejrzyste przedstawienie rozkładu wyników uzyskanych przez studentów z różnych przedmiotów. W projekcie histogram ukazuje, ile razy każda ocena pojawia się w zbiorze danych dla poszczególnych przedmiotów, co pozwala na intuicyjne zrozumienie rozkładu ocen w analizowanej grupie.



Rysunek 3: Tabela podsumowująca ilości ocen

```
1  const ocenyToHistogram = function (oceny)
2  {
3    const hist = {};
4    for (let i = 2.0; i <= 5.0; i+= 0.5) {
5      if (i == 2.5) {
6        continue;
7      }
8      hist[`o${i}`] = 0;
9    }
10
11   for (const ocena of oceny) {
12     hist[`o${ocena}`] += 1;
13   }
14
15   return Object.values(hist);
16 }
17
```

Listing 3: Funkcja tworząca histogram ze zbioru ocen



Rysunek 4: Histogram ocen

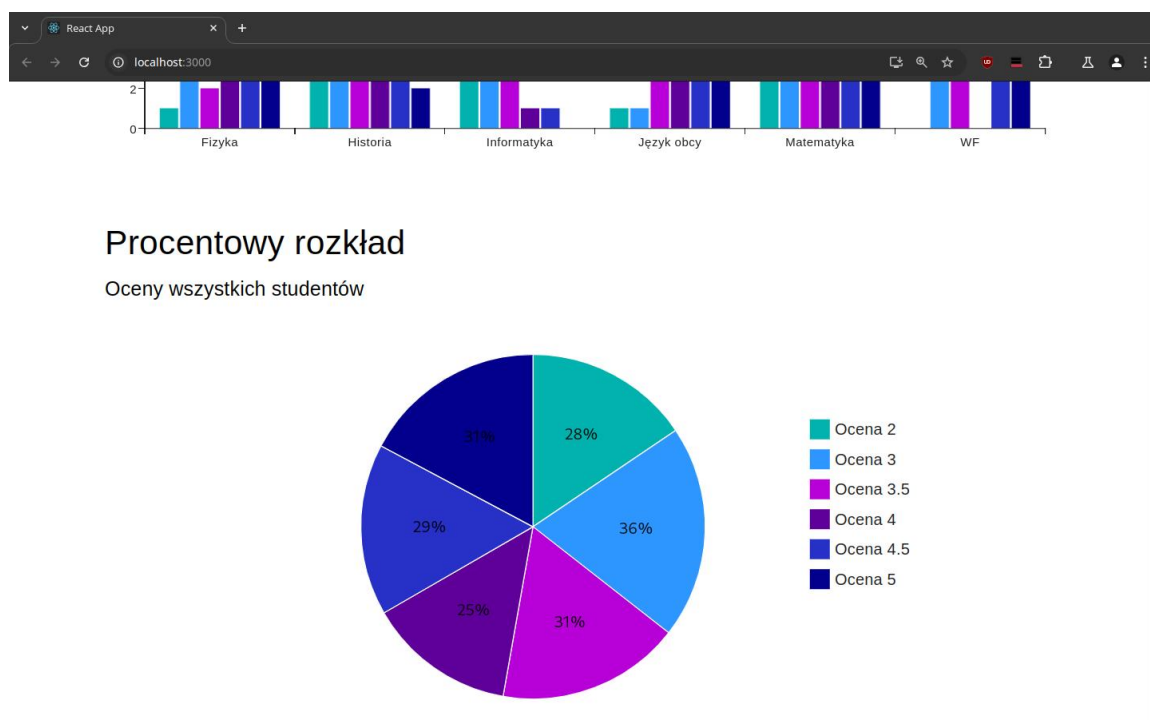
Dzięki powyższym informacjom, możemy również przeanalizować ogólną ilość poszczególnych stopni. Przedstawimy to na wykresie kołowym:

```

1 <PieChart
2   width={700}
3   height={400}
4   series={[{
5     arcLabel: (item) => `${item.value}%`,
6     arcLabelMinAngle: 35,
7     arcLabelRadius: '60%',
8     data: histogram.filter(n => n.przedmiot == "Wszystkie").at(0).hist.map((n, i)
9       => {
10        return {
11          id: i,
12          value: n,
13          label: `Ocena ${2.0 + i * 0.5 + (i >= 1 ? 0.5 : 0)}%`
14        }
15      })
16    ]}]
17 />

```

Listing 4: Generowanie wykresu kołowego



Rysunek 5: Histogram ocen

2.5 Miary tendencji

Miary tendencji centralnej i zmienności są kluczowymi wskaźnikami w analizie danych statystycznych, pozwalającymi określić podstawowe właściwości rozkładu ocen. W projekcie wyznaczono następujące wskaźniki: średnią arytmetyczną, medianę, dominantę, wariancję oraz odchylenie standardowe. Każda z tych miar dostarcza cennych informacji o danych i pozwala na lepsze zrozumienie wyników studentów.

- **Średnia arytmetyczna:** Średnia arytmetyczna (oznaczana jako \bar{X}) to suma wszystkich wartości w zbiorze, podzielona przez liczbę tych wartości. Średnia arytmetyczna jest podstawową miarą tendencji centralnej.

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

gdzie:

- X_i – wartość pojedynczej obserwacji,
 - n – liczba obserwacji.
- **Mediana:** Mediana to środkowa wartość zbioru uporządkowanych danych. W przypadku liczby nieparzystej jest to wartość dokładnie w środku, a przy liczbie parzystej – średnia arytmetyczna dwóch środkowych wartości.
 - **Dominanta:** Dominanta, czyli wartość modalna, to wartość występująca najczęściej w zbiorze danych. Jest to prosta miara, która pozwala zidentyfikować najczęściej uzyskiwaną ocenę.

- **Wariancja:** Wariancja (oznaczana jako σ^2) jest miarą rozproszenia danych wokół średniej. Oblicza się ją jako średnią kwadratów odchyłeń poszczególnych wartości od średniej.

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$$

- **Odchylenie standardowe:** Odchylenie standardowe (σ) to pierwiastek kwadratowy z wariancji. Jest intuicyjną miarą rozproszenia, wyrażoną w tych samych jednostkach co wartości w zbiorze danych.

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2}$$

Te miary pozwalają uzyskać szczegółowy wgląd w rozkład ocen studentów, w tym średnią wartość wyników, częstość występowania ocen, a także stopień ich zróżnicowania.

```

1  const getMedian = function (oceny)
2  {
3    const sorted = oceny.sort();
4
5    if (sorted % 2 !== 0)
6      return sorted[Math.floor(sorted.length / 2)];
7
8    const half = sorted.length / 2;
9    return (oceny[half] + oceny[half - 1]) / 2;
10 }
11
12 const getDominion = function (oceny)
13 {
14   const counts = {};
15   for (let i = 2.0; i <= 5.0; i += 0.5) {
16     counts[`${i}`] = 0;
17   }
18
19   for (let ocena of oceny) {
20     counts[`${ocena}`] += 1;
21   }
22
23   const domain = [];
24   const most = Object.values(counts).reduce((a, b) => Math.max(a, b));
25   for (const [k, v] of Object.entries(counts)) {
26     if (v == most) {
27       domain.push(k.substring(1));

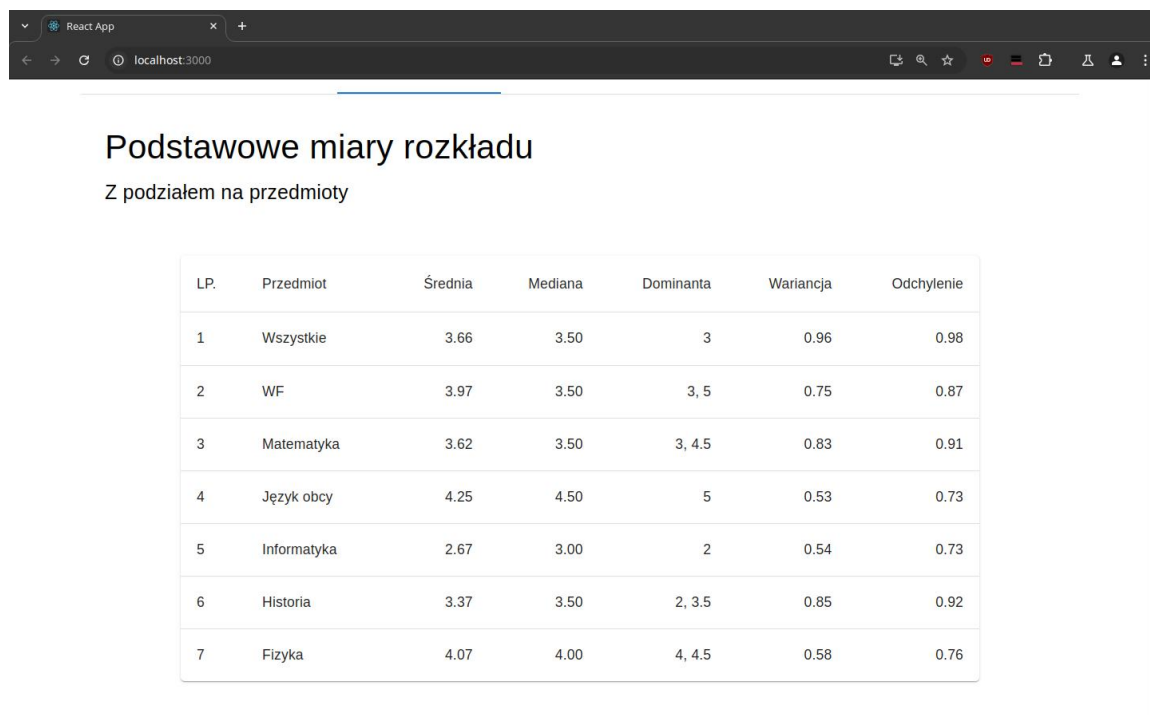
```

```

28     }
29 }
30
31 return domain.join(", ");
32 }
33
34 const prepMeasures = function (oceny)
35 {
36     const res = [];
37     const avg = oceny.reduce((a, b) => a + b) / oceny.length;
38     res.push(avg.toFixed(2));
39     res.push(getMedian(oceny).toFixed(2));
40     res.push(getDominion(oceny));
41
42     const war = oceny.map(n => Math.pow(n - avg, 2)).reduce((a, b) => a + b) /
        ↪ oceny.length;
43     res.push(war.toFixed(2));
44     res.push(Math.sqrt(war).toFixed(2));
45
46     return res;
47 }

```

Listing 5: Programistyczne podejście do matematycznych obliczeń



LP.	Przedmiot	Średnia	Mediana	Dominanta	Wariancja	Odchylenie
1	Wszystkie	3.66	3.50	3	0.96	0.98
2	WF	3.97	3.50	3, 5	0.75	0.87
3	Matematyka	3.62	3.50	3, 4, 5	0.83	0.91
4	Język obcy	4.25	4.50	5	0.53	0.73
5	Informatyka	2.67	3.00	2	0.54	0.73
6	Historia	3.37	3.50	2, 3, 5	0.85	0.92
7	Fizyka	4.07	4.00	4, 4, 5	0.58	0.76

Rysunek 6: Tabela podsumowująca wyniki

2.6 Korelacja Pearsona

Współczynnik korelacji Pearsona jest miarą liniowej zależności pomiędzy dwiema zmiennymi. Wartość współczynnika korelacji przyjmuje wartości od -1 do 1 . Współczynnik ten pozwala ocenić, czy wzrost jednej zmiennej wiąże się ze wzrostem lub spadkiem drugiej zmiennej oraz w jakim stopniu. Interpretacja współczynnika korelacji jest następująca:

- Wartość $r = 1$ oznacza doskonałą dodatnią korelację liniową.
- Wartość $r = -1$ oznacza doskonałą ujemną korelację liniową.
- Wartość $r = 0$ wskazuje na brak korelacji liniowej między zmiennymi.

Współczynnik korelacji Pearsona r jest obliczany według wzoru:

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \cdot \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

gdzie:

- X_i i Y_i – wartości dwóch zmiennych dla i -tego przypadku,
- \bar{X} i \bar{Y} – średnie arytmetyczne zmiennych X i Y ,
- n – liczba obserwacji.

Wartość współczynnika korelacji Pearsona w analizie danych z ocenami pozwala na identyfikację zależności między wynikami z różnych przedmiotów. Dzięki temu można określić, czy istnieje statystycznie istotna zależność pomiędzy ocenami z dwóch różnych przedmiotów, co może pomóc w zrozumieniu, czy osiągnięcia w jednym przedmiocie mają tendencję do wzrostu lub spadku wraz z wynikami w innym.

Współczynniki korelacji Pearsona

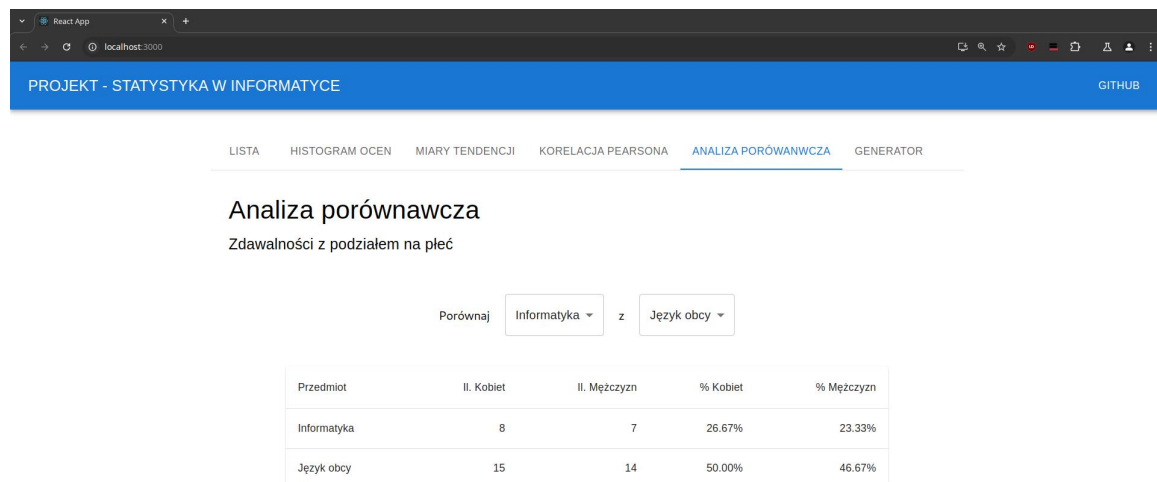
Dla par przedmiotów

	Fizyka	Historia	Informatyka	Język obcy	Matematyka	WF
Fizyka	-	0.14	0.14	-0.12	0.54	-0.28
Historia	0.14	-	-0.15	-0.17	0.19	-0.15
Informatyka	0.14	-0.15	-	0.08	0.20	0.27
Język obcy	-0.12	-0.17	0.08	-	-0.28	-0.21
Matematyka	0.54	0.19	0.20	-0.28	-	0.02
WF	-0.28	-0.15	0.27	-0.21	0.02	-

Rysunek 7: Tabela podsumowująca wyniki

2.7 Analiza porównawcza

Aplikacja umożliwia porównanie zdawalności z poszczególnych przedmiotów z podziałem na płeć studenta.



Rysunek 8: Tabela podsumowująca wyniki

3. Instrukcja uruchomienia aplikacji

Aby uruchomić aplikację stworzoną w React, należy wykonać kilka kroków konfiguracyjnych:

1. Zainstaluj Node.js i npm

Upewnij się, że na komputerze zainstalowane są **Node.js** oraz **npm** (Node Package Manager). Node.js jest niezbędny do uruchamiania aplikacji w React, a npm służy do zarządzania bibliotekami.

Aby sprawdzić, czy są zainstalowane, uruchom następujące polecenia w terminalu:

```
node -v
npm -v
```

Jeśli nie masz ich zainstalowanych, pobierz odpowiednie wersje z <https://nodejs.org/>.

2. Sklonuj repozytorium z kodem aplikacji

Kod znajduje się w repozytorium na GitHubie, sklonuj go lokalnie za pomocą polecenia:

```
git clone https://github.com/quakcin/Statystyka-w-informatyce-Projekt
```

Następnie przejdź do katalogu projektu:

```
cd Statystyka-w-informatyce-Projekt/uusos
```

3. Zainstaluj zależności

W katalogu projektu uruchom następujące polecenie, aby zainstalować wszystkie wymagane biblioteki:

```
npm install
```

4. Uruchom aplikację w trybie deweloperskim

Po zakończeniu instalacji zależności uruchom aplikację w trybie deweloperskim za pomocą komendy:

```
npm start
```

Aplikacja zostanie uruchomiona, a domyślnie będzie dostępna pod adresem `http://localhost:3000`. Po wprowadzeniu zmian w kodzie aplikacja automatycznie się przeładuje.

4. Podsumowanie

W ramach projektu "Analiza wyników egzaminów studentów z różnych przedmiotów" zaprezentowano narzędzie do analizy i wizualizacji ocen studentów. Aplikacja została oparta na bibliotece React, a dane o ocenach studentów były generowane przy pomocy autorskiego generatora, co pozwoliło na uzyskanie różnorodności ocen. Wszystkie dane zostały przechowywane w bazie Dexie.js, która zapewniała wygodny dostęp do danych bez potrzeby implementacji backendu.

W projekcie zostały obliczone podstawowe miary statystyczne, takie jak średnia, mediana, dominanta, wariancja, odchylenie standardowe oraz współczynnik korelacji Pearsona. Te miary umożliwiają analizę rozkładu wyników w poszczególnych przedmiotach oraz między przedmiotami, dostarczając cennych informacji o osiągnięciach studentów.

Aplikacja umożliwia także wizualizację danych za pomocą wykresów, w tym histogramu ocen, co pozwala na łatwe zrozumienie rozkładu wyników. Interfejs aplikacji jest przyjazny dla użytkownika, dzięki zastosowaniu biblioteki MUI.

Projekt umożliwia przeprowadzenie szczegółowej analizy wyników studentów. Dzięki elastyczności aplikacji, możliwe jest dostosowanie jej do różnych wymagań analitycznych i rozwoju w przyszłości.