Aplikacja do identyfikacji płci mózgu

Projekt Indywidualny: Mikołaj Sęklewski 282608

1.Wstęp

Celem projektu było stworzenie na stronie internetowej aplikacji służącej do identyfikacji płci mózgu. Gotowy test znalazłem w internecie, a pochodzi on z książki „Płeć Mózgu” autorstwa A.Moir, D.Jessel. Moim zadaniem było zaimplementowanie go jako aplikacji internetowej za pomocą React.js.

2. Środowisko

Aplikacja tworzona była w technologii React.js, czyli bibliotece języka programowania JavaScript.

Jako edytora tekstu używałem Visual Studio Code z opcją kolorowania składni.

Istotnym elementem było również środowisko uruchomieniowe Node.js, które zapewniło narzędzia potrzebne do tworzenia stron lokalnie na komputerze.

Do inspekcji i testowania tworzonej aplikacji użyłem przeglądarki Google Chrome z zainstalowanym dodatkiem React Developer Tools.

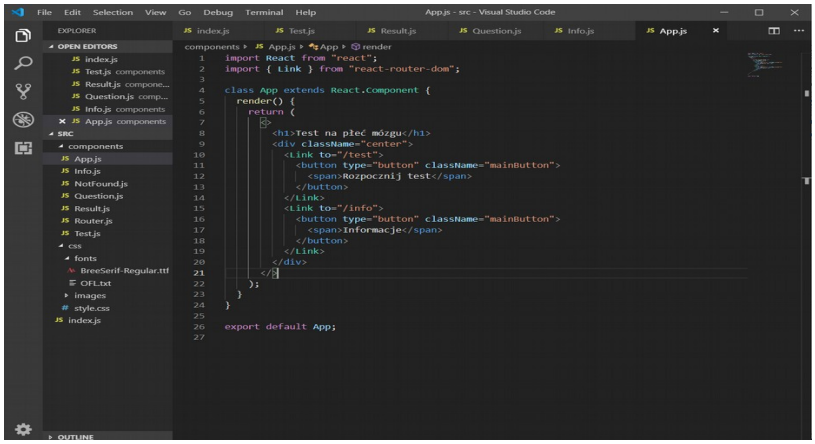


Figure 1: Visual Studio Code

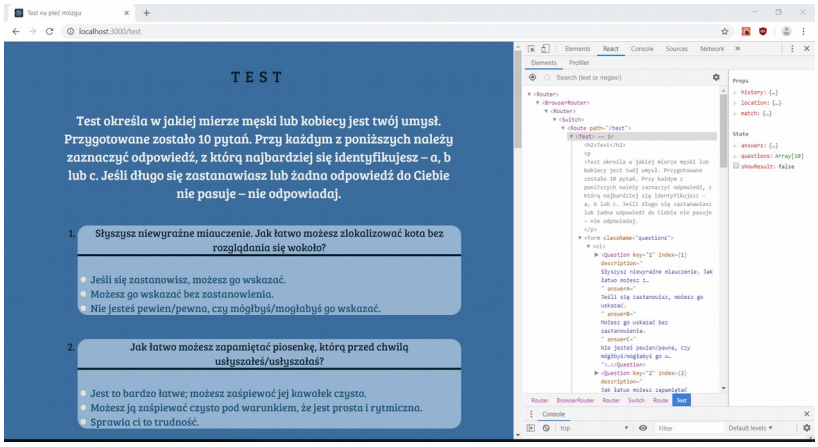


Figure 2: React Developer Tools

3. Realizacja

Struktura

Pierwszym krokiem było zaprojektowanie ogólnej struktury aplikacji. Podzieliłem aplikacje na 5 podstawowych podstron oraz ustaliłem sposób poruszanie się między nimi.

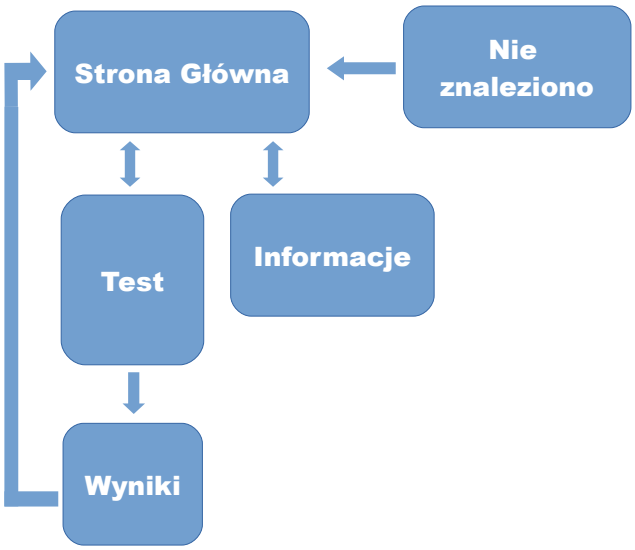
1) Strona główna

2) Informacje

3) Test

4) Wyniki

5) Nie znaleziono (wyświetlana gdy zostanie wpisany błędny adres)



Komponenty Reacta odpowiadają zaplanowanej strukturze, rozszerzając je o komponent zajmujący się trasowaniem<Router> oraz wydzielony komponent na pojedyncze pytania <Question>.

Kod

Plik ‘index.js’ renderuje stronę na podstawie tego co dostarczy <Router> oraz importuje plik CSS odpowiedzialny za wygląd strony.

index.js

```
1  import React from "react";
2  import { render } from "react-dom";  101.5K (gzipped: 32.9K)
3  import Router from "./components/Router";
4  import "./css/style.css";
5
6  render(<Router />, document.querySelector("#main"));
7
```

Komponent <Router> przekierowuje do innych komponentów w zależności od wpisanego adresu.

```
9  const Router = () => (
10   <BrowserRouter>
11     <Switch>
12       <Route exact path="/" component={App} />
13       <Route path="/test" component={Test} />
14       <Route path="/info" component={Info} />
15       <Route path="/result" component={Result} />
16       <Route component={NotFound} />
17     </Switch>
18   </BrowserRouter>
19 );
```

Strona główna renderowana jest w komponencie <App>, wyświetla nazwę aplikacji oraz przyciski linkujące do kolejnych podstron.

```
class App extends React.Component {
  render() {
    return (
      <>
        <h1>Test na płeć mózgu</h1>
        <div className="center">
          <Link to="/test">
            <button type="button" className="mainButton">
              <span>Rozpocznij test</span>
            </button>
          </Link>
          <Link to="/info">
            <button type="button" className="mainButton">
              <span>Informacje</span>
            </button>
          </Link>
        </div>
      </>
    );
  }
}
```

Informacje o teście znajdują się w komponencie <Info>

```
const Info = () => (
  <>
    <div>
      <h2>Informacje</h2>
      <p>
        Test określa w jakiej mierze męski lub kobiecy jest twój umysł. To, w
        jakim stopniu zachowanie kobiet i mężczyzn jest kobiece lub męskie,
        zależy od modelu, według którego uformowany jest ich mózg. Hormony
        płciowe wywierają wpływ na kształtowanie mózgu w rozwoju płodowym
        powodując powstawanie podstawowych różnic między ludźmi.
        <br />
        Test został opartyna podstawie książki "Płeć mózgu" -A.Moir, D.Jessel .
        <br />
        Autor aplikacji -Mikołaj Sęklewski
      </p>
    </div>
    <div className="center">
      <Link to="/">
        <button type="button">
          <span>Strona główna</span>
        </button>
      </Link>
    </div>
  </>
);
```

Największy komponent <Test> przechowuje bazę pytań, które są następnie renderowane jako lista komponentów <Question>.

```
class Test extends React.Component {
  state = {
    questions: [
      {
        id: 1,
        description:
          "Słyszysz niewyraźne miauczenie. Jak łatwo możesz zlokalizować kota bez rozgląd
        answerA: "Jeśli się zastanowisz, możesz go wskazać.",
        answerB: "Możesz go wskazać bez zastanowienia.",
        answerC: "Nie jesteś pewien/pewna, czy mógłbyś/mogłabyś go wskazać."
      },
      {
        id: 2,
        description:
          " Jak łatwo możesz zapamiętać piosenkę, którą przed chwilą usłyszałeś/usłyszała
```

```
<form className="questions" onSubmit={this.showResult}>
  <ol>
    {this.state.questions.map(question => (
      <Question
        key={question.id}
        index={question.id}
        description={question.description}
        answerA={question.answerA}
        answerB={question.answerB}
        answerC={question.answerC}
        setAnswer={this.getAnswer}
      />
    ))}
  </ol>
  <div className="center">
    <button type="submit">
      <span>Wynik</span>
    </button>
  </div>
</form>
```

Na bieżąco uaktualniania jest lista zaznaczonych odpowiedzi.

```
getAnswer = (answer, n) => {
  const answers = { ...this.state.answers };
  answers[n] = answer;
  this.setState({ answers });
};
```

Po zatwierdzeniu odpowiedzi przyciskiem „Wynik” test zostaje ukryty i zastąpiony wynikiem.

```
  showResult = event => {
    event.preventDefault();
    this.setState({ showResult: true });
  };
```

Wyniki renderowane są w komponencie <Result> do którego przekazane zostały odpowiedzi z testu.

```
          <Result answers={this.state.answers} />
```

Po wybraniu naszej płci zostaje policzona liczba odpowiedzi A, B,C oraz pustych. Następnie w zależności od płci liczony jest ostateczny wynik do wyświetlenia.

```
  render() {
    return (
      <>
        <div className="question">
          <h3>Zaznacz swoją płeć:</h3>
          <div className="result">
            <label>
              <input
                type="radio"
                name="sex"
                value="female"
                onChange={this.calcResult}
              />
              {"Kobieta "}
            </label>
            <label>
              <input
                type="radio"
                name="sex"
                value="male"
                onChange={this.calcResult}
              />
              {"Mężczyzna "}
            </label>
          </div>
        </div>
        <h2>Twój wynik: {this.state.showScore ? this.state.score : ""}</h2>
```

```
calcResult = event => {
  const answers = Object.values(this.props.answers);
  const countA = answers.filter(answer => answer === "A").length;
  const countB = answers.filter(answer => answer === "B").length;
  const countC = answers.filter(answer => answer === "C").length;
  const countEmpty = 10 - (countA + countB + countC);

  var sum = 0;
  switch (event.target.value) {
    case "female":
      sum = (countB + countEmpty) * 5 + countC * -5 + countA * 15;
      break;

    case "male":
      sum = (countB + countEmpty) * 5 + countC * -5 + countA * 10;
      break;

    default:
  }

  this.setState({ score: sum });
  this.setState({ showScore: true });
};
```

<Question> odpowiada za renderowanie pojedynczego pytania oraz przesyłanie swojemu rodzicowi <Test> zaznaczonej odpowiedzi.

```
render() {
  return (
    <li className="question" >
      <div>
        <h3>{this.props.description}</h3>
        <label>
          <input
            type="radio"
            name={"question" + this.props.index}
            value="A"
            onChange={this.handleChange}
          />{" "}
          {this.props.answerA}
        </label>
        <br />
        <label>
          <input
            type="radio"
            name={"question" + this.props.index}
            value="B"
```

```
handleChange = event => {
  this.props.setAnswer(event.target.value, this.props.index);
};
```

W przypadku wpisania w pasku przeglądarki błędnego adresu, zostajemy przekierowani do komponentu <NotFound> który umożliwia powrót do strony głównej.

```jsx
const NotFound = () => (
  <>
    <h2>Not Found!</h2>
    <div className="center">
      <Link to="/">
        <button type="button">
          <span>Strona główna</span>
        </button>
      </Link>
    </div>
  </>
);
```

Plik ‘style.css’ odpowiada za wygląd strony, ustawiając czcionkę, kolory oraz kształt i animacje elementów na stronie.

```
css > # style.css > ...
```

```css
body
{
  background-color: #356799;
  font-family: 'MyFont','serif', sans-serif;
  font-size: 20px;
  color: #efefef;
}

@font-face {
  font-family: 'MyFont';
  src:url("./fonts/BreeSerif-Regular.ttf");
}

.center {
  display: flex;
```

Podgląd Strony

W pliku ‘package.json’ znajdują się deklaracje modułów umożliwiających lokalne tworzenie oraz podgląd strony. Po pobraniu modułów poleceniem „npm install”, uruchamiam aplikacje poleceniem „npm start”.

```
E:\NAUKA\Projekt Indywidualny - Test na płeć mózgu
λ npm install
[..................] | preinstall:cotd: info lifecycle cotd@0.0.3~preinstall: cotd@0.0.3
```

```
E:\NAUKA\Projekt Indywidualny - Test na płeć mózgu
λ npm start

> cotd@0.0.3 start E:\NAUKA\Projekt Indywidualny - Test na płeć mózgu
> react-scripts start

Compiled successfully!

You can now view cotd in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://192.168.56.1:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.
```

Gotowa Aplikacja

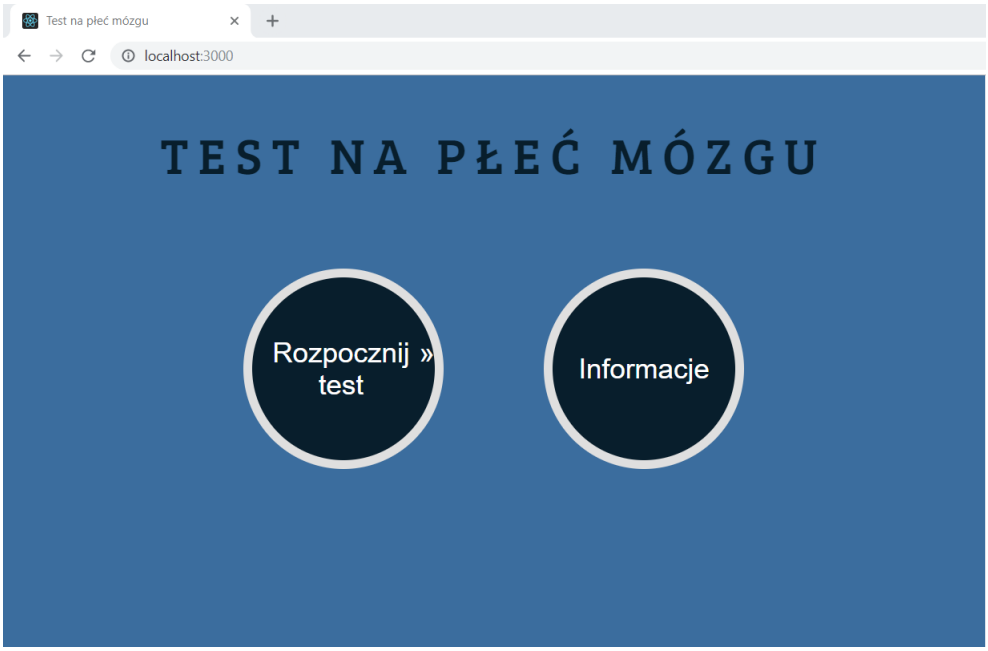


Figure 3: Strona główna

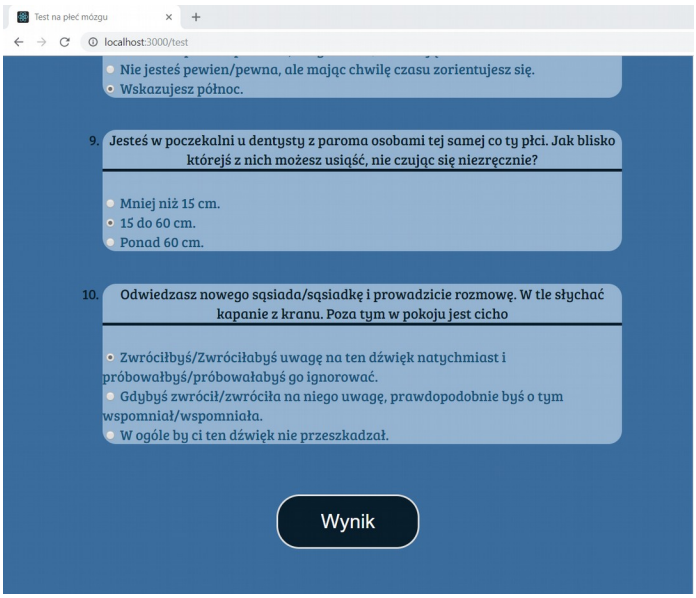


Figure 4: Test

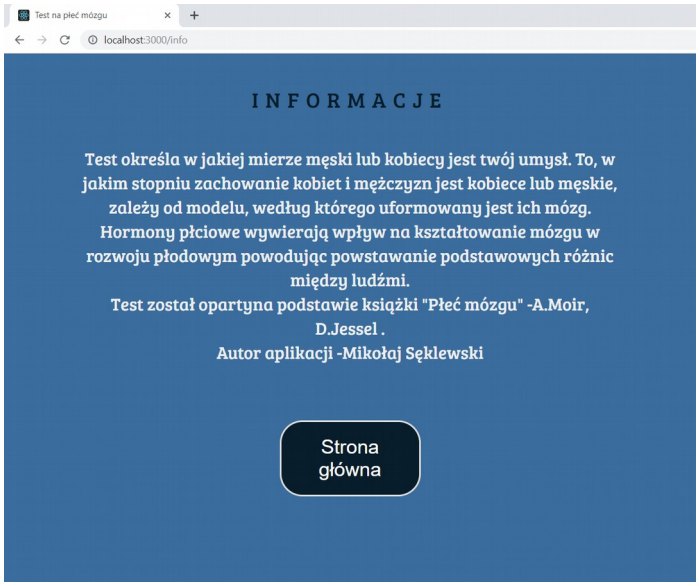


Figure 5: Informacje

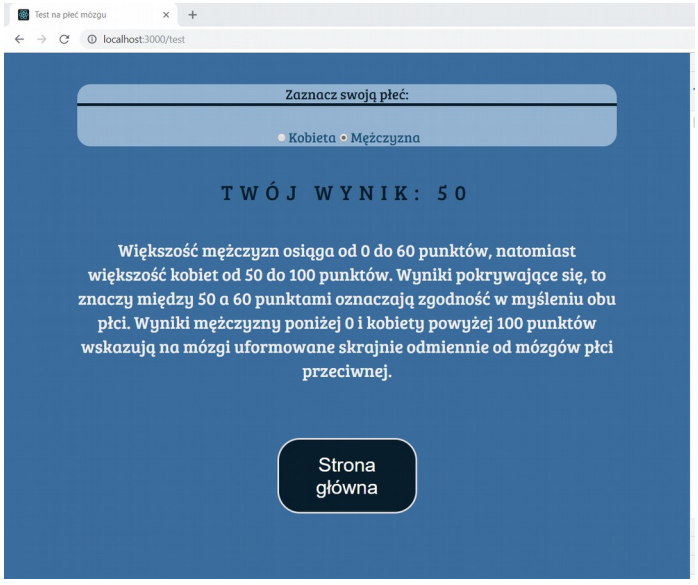


Figure 6: Wyniki testu

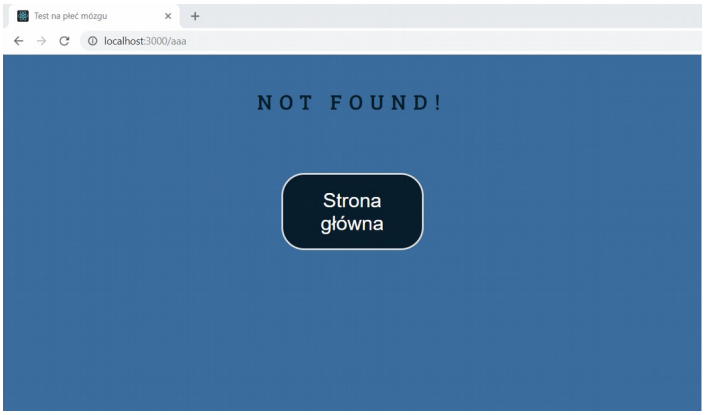


Figure 7: Błędny adres strony

4. Wnioski

Realizacja tego projektu pozwoliła mi na nauczenie się podstaw tworzenia aplikacji webowych oraz poznanie narzędzi do tego służących. Technologia React.js jest dość prosta do nauki i intuicyjna, głównie dzięki zamknięciu wszystkiego w komponenty oraz zastosowaniu jednokierunkowego przepływu danych(od rodzica do dziecka). Wadą może być to że za pomocą dodatku z narzędziami developerskimi do przeglądarki Chrome, każdy ma dostęp do potencjalnie wrażliwych danych.

Możliwe perspektywy rozwoju aplikacji to stworzenie oddzielnej bazy pytań, która umożliwi łatwe rozbudowywanie oraz selekcje konkretnych pytań do testu.