Generator dokumentu PDF planu treningowego

**Spis treści**

1. Wstęp .......................................................... 2
2. Cel projektu .................................................... 2
3. Technologia ..................................................... 2
4. Projekt systemu .............................................. 2
   * Plik wejściowy ............................................ 2
   * Moduł ....................................................... 3
   * Moduł Planów Treningowych ........................ 3
   * Moduł Wyświetlania ................................... 3
   * Moduł Konwersji ........................................ 4
   * Integracja modułów ................................... 4
   * Plik wynikowy ………………………………………. 5
5. Realizacja Aplikacji ....................................... 5
   * Obsługa programu ....................................... 5
   * Schemat budowy aplikacji ........................... 9
   * Użycie aplikacji ........................................ 9
6. Sposób rozbudowy projektu ............................ 10
7. Propozycje rozbudowy projektu ....................... 10
8. Literatura ...................................................... 10
9. **Wstęp**

Tworzenie aplikacji webowej "Generator dokumentu PDF planu treningowego" opiera się na przekształceniu planów treningowych w formacie Word na dokumenty PDF. Użytkownicy mogą logować się, dodawać swoje plany treningowe poprzez formularz oraz pobierać plany treningowe w formacie PDF. Aplikacja umożliwia przypisanie planów do trzech grup: początkujący, średniozaawansowany i zaawansowany.

1. **Cel projektu**

Celem projektu jest zrealizowanie aplikacji umożliwiającej przekształcenie planów treningowych zapisanych w formacie Word na dokumenty PDF, które można pobrać. Użytkownik może wgrać swój plan, przypisując go do odpowiedniej grupy treningowej, a inni użytkownicy mogą wyświetlać i pobierać te plany.

1. **Technologia**

Ze względu na wieloplatformowość oraz graficzny interfejs zastosowano język Python i framework Django oraz Javascript i framework React.js, a do konwersji dokumentów Word na PDF użyto bibliotek ReportLab i python-docx. Do przechowywania danych takich jak zarejestrowani użytkownicy i pliki planów treningowych w formacie Word użyto bazy danych PostgreSQL.

1. **Projekt systemu**

**Organizacja działania:**

* Użytkownicy logują się do aplikacji.
* Dodawanie planów treningowych w formacie Word.
* Przypisywanie planów do odpowiednich grup treningowych.
* Wyświetlanie i pobieranie planów treningowych w formacie PDF.

**Plik wejściowy**

Plik wejściowy, który użytkownik może przesłać do systemu powinien być dokumentem w formacie Microsoft Word (.docx). Każdy plik powinien zawierać plan treningowy, który użytkownik chce zamienić na format PDF. System akceptuje pliki w trzech kategoriach treningowych:

* Początkujący
* Średniozaawansowany
* Zaawansowany

Dokument Word może zawierać tekst sformatowany według preferencji użytkownika, włącznie z listami, nagłówkami oraz zwykłym tekstem. Wszystkie elementy tekstowe z dokumentu są konwertowane na format PDF z zachowaniem podstawowego formatowania.

**Moduł**

Moduły projektu podzielone są na kilka głównych części:

* **Moduł Planów Treningowych**: Zarządza dodawaniem i przechowywaniem planów treningowych.
* **Moduł Wyświetlania**: Umożliwia wyświetlanie planów treningowych.
* **Moduł Konwersji**: Konwertuje pliki Word na PDF.

**Moduł Planów Treningowych**

Odpowiada za dodawanie, przechowywanie i organizację planów treningowych. Plany treningowe są przechowywane w bazie danych.

class TrainingPlanUploadView(views.APIView):

    permission\_classes = [IsAuthenticated]

    parser\_classes = (MultiPartParser, FormParser)

    def post(self, request, \*args, \*\*kwargs):

        serializer = TrainingPlanSerializer(data=request.data)

        if serializer.is\_valid():

            serializer.save()

            return Response(serializer.data, status=status.HTTP\_201\_CREATED)

        else:

            return Response(serializer.errors, status=status.HTTP\_400\_BAD\_REQUEST)

parser\_classes = (MultiPartParser, FormParser) – ta linijka kodu umożliwia przetwarzanie danych w formacie wieloczęściowym, co jest konieczne dla przesyłania plików (takich jak pliki Word)

**Moduł Wyświetlania**

Umożliwia użytkownikom przeglądanie planów treningowych oraz pobieranie ich w formacie PDF.

class TrainingPlanView(views.APIView):

    permission\_classes = [IsAuthenticated]

    def get(self, request, plan\_type):

        try:

            plan = TrainingPlan.objects.get(plan\_type=plan\_type)

            doc = Document(plan.document)

            full\_text = [paragraph.text for paragraph in doc.paragraphs]

            return Response({'plan': "\n".join(full\_text)})

        except TrainingPlan.DoesNotExist:

            return Response({'error': 'Plan not found'}, status=status.HTTP\_404\_NOT\_FOUND)

        except Exception as e:

            return Response({'error': str(e)}, status=status.HTTP\_500\_INTERNAL\_SERVER\_ERROR)

**Moduł Konwersji**

Odpowiada za konwersję plików Word na PDF. Wykorzystuje biblioteki ReportLab i python-docx do generowania PDF-ów.

class TrainingPlanPDFView(views.APIView):

    permission\_classes = [IsAuthenticated]

    def get(self, request, plan\_type):

        try:

            plan = TrainingPlan.objects.get(plan\_type=plan\_type)

            response = HttpResponse(content\_type='application/pdf')

            response['Content-Disposition'] = f'attachment; filename="{plan\_type}\_training\_plan.pdf"'

            p = canvas.Canvas(response, pagesize=letter)

            doc = Document(plan.document)

            text = p.beginText(40, 750)

            text.setFont('Times-Roman', 12)

            for paragraph in doc.paragraphs:

                text.textLine(paragraph.text)

            p.drawText(text)

            p.showPage()

            p.save()

            return response

        except TrainingPlan.DoesNotExist:

            return Response({'error': 'Plan not found'}, status=status.HTTP\_404\_NOT\_FOUND)

        except Exception as e:

            return Response({'error': str(e)}, status=status.HTTP\_500\_INTERNAL\_SERVER\_ERROR)

**Integracja modułów**

Integracja modułów polega na wzajemnym komunikowaniu się pomiędzy modułami za pomocą API Django. Moduł dodawania planów treningowych zapisuje dane w bazie, a moduł konwersji przekształca je na pliki PDF, które następnie mogą być wyświetlane przez moduł wyświetlania.

**Plik Wynikowy**

Plik wynikowy generowany przez system to dokument w formacie PDF, który jest cyfrową reprezentacją planu treningowego zawartego w przesłanym dokumencie Word. Plik PDF jest generowany przy użyciu biblioteki ReportLab, która pozwala na precyzyjne odwzorowanie układu tekstu.

Każdy wygenerowany PDF jest formatowany z zachowaniem czytelności i estetyki:

* Rozmiar strony: List (8.5 x 11 cali)
* Czcionka: Times New Roman, rozmiar 12 pt
* Marginesy: Standardowe marginesy 1 cal z każdej strony

Użytkownicy mogą pobierać pliki PDF bezpośrednio przez interfejs użytkownika, klikając odpowiedni przycisk, który inicjuje pobieranie pliku na urządzenie użytkownika.

1. **Realizacja Aplikacji**

**Obsługa programu**

Aplikacja umożliwia użytkownikom logowanie się, dodawanie planów treningowych oraz przeglądanie i pobieranie planów treningowych w formacie PDF. Użytkownicy są autoryzowani za pomocą tokenów JWT.

* Rejestracja użytkownika

Obraz zawierający tekst, zrzut ekranu, Czcionka, Grafika

Opis wygenerowany automatycznie

* Logowanie użytkownika

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

* Widok planu treningowego w Wordzie

Obraz zawierający tekst, oprogramowanie, Ikona komputerowa, Strona internetowa

Opis wygenerowany automatycznie

* Strona główna aplikacji

Przyciski „Początkujący, Średniozaawansowany i Zaawansowany odpowiadają za wyświetlenie zawartości dodanego poprzez formularz pliku Worda. Przyciski Pobierz PDF dla początkujących, średniozaawansowanych i zaawansowanych odpowiadają za konwersję danych z pliku Word do PDF i umożliwiają pobranie tego pliku PDF użytkownikowi. Formularz pod przyciskami umożliwia dodanie pliku w formacie Word oraz wybór kategorii zaawansowania planu treningowego.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Obraz zawierający tekst, oprogramowanie, Ikona komputerowa, Strona internetowa

Opis wygenerowany automatycznie

* Pomyślne dodanie planu treningowego

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

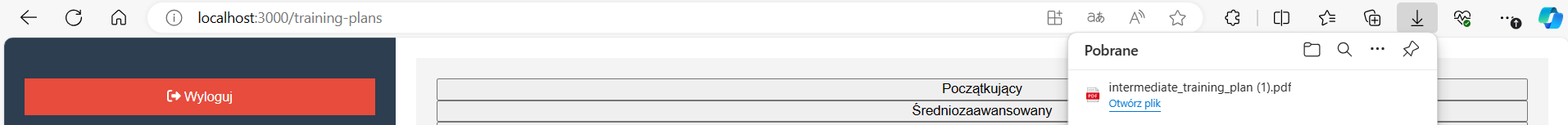
Opis wygenerowany automatycznie

* Wyświetlanie zawartości pliku Word

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

* Pobranie pliku PDF przez użytkownika



* Zawartość pliku PDF

Obraz zawierający tekst, oprogramowanie, Ikona komputerowa, Strona internetowa

Opis wygenerowany automatycznie

* Zapisane plany treningowe w bazie danych

Obraz zawierający tekst, oprogramowanie, numer, Ikona komputerowa

Opis wygenerowany automatycznie

**Schemat budowy aplikacji**

Główna część aplikacji to funkcje obsługujące dodawanie, przeglądanie i konwersję planów treningowych:

* **Odczyt danych**: Pobiera plik Word od użytkownika.
* **Przetwarzanie danych**: Konwertuje plik Word na PDF.
* **Zapis danych**: Przechowuje pliki Word w bazie danych.

**Użycie aplikacji**

1. Użytkownik loguje się do systemu.
2. Dodaje plan treningowy, przypisując go do odpowiedniej grupy treningowej.
3. Użytkownik może wyświetlić plan treningowy i pobrać go w formacie PDF.
4. **Sposób rozbudowy projektu**

Aby rozbudować projekt, można dodać nowe funkcje, takie jak:

* Personalizacja planów treningowych.
* Możliwość edycji istniejących planów.
* Integracja z innymi narzędziami do zarządzania treningiem.

1. **Propozycje rozbudowy projektu**

* Dodanie interfejsu użytkownika dla lepszej interakcji.
* Implementacja dodatkowych formatów dokumentów.
* Ulepszona autoryzacja i uwierzytelnianie użytkowników.

1. **Literatura**

* Python i Django. Programowanie aplikacji webowych (Autorzy: Jeff Forcier, Paul Bissex, Wesley Chun)
* https://docs.reportlab.com/
* https://python-docx.readthedocs.io/en/latest/