

Rysunek Logo aplikacji

Learn-ey

Spis treści

[Opis celu projektu 2](#_Toc148606307)

[Wymagania funkcjonalne 2](#_Toc148606308)

[Wymagania niefunkcjonalne 3](#_Toc148606309)

[Opis wybranego środowiska programistycznego 3](#_Toc148606310)

[User stories 5](#_Toc148606311)

[Repozytorium zdalne 6](#_Toc148606312)

[Product backlog 7](#_Toc148606313)

# Opis celu projektu

Learn-ey to aplikacja desktopowa wspomagająca proces nauki. Celem projektu jest stworzenie narzędzia, które pozwoli użytkownikowi na skuteczniejsze przyswajanie wiedzy z wybranej przez siebie dziedziny lub słówek w obcych językach. Program będzie charakteryzował się prostotą użytkowania   
i minimalistycznym interfejsem. Podczas korzystania z aplikacji użytkownik będzie miał możliwość wprowadzić do aplikacji materiał, który chce przyswoić za pomocą formularza lub za pomocą wcześniej przygotowanego pliku, z którego aplikacja wczyta materiały. Nauka odbywać się będzie w oparciu   
o metodę pomodoro - sposób zarządzania czasem ukierunkowany na zwiększenie produktywności oraz koncentracji. Polega on na rozbiciu procesu nauki na sekcje: 25 min intensywnego przyswajania wiedzy i odpoczynek przez kolejne 5 min. Następnie należy powtórzyć cały proces. Po 4 pełnych cyklach następuje dłuższa przerwa zazwyczaj 15 do 30 minut.

Zadania, poprzez które następuje nauka są technikami aktywizującymi - zróżnicowanie typu zadań w celu aktywowania większej ilości funkcji poznawczych oraz w celu zredukowania schematycznego myślenia, co może istotnie wpłynąć na efektywność procesu zapamiętywania treści.

Zadania jakie będą prezentowane użytkownikowi to:

1. Gra w wisielca
2. Fiszki
3. Prawda/Fałsz
4. Wskaż poprawną definicję
5. Wskaż poprawne słowo
6. Dopasuj definicję do zagadnienia

Użytkownik po każdej odbytej sesji nauki otrzyma raport z wygenerowany na podstawie uzyskanych przez program odpowiedzi.

Projekt napisany zostanie w języku Python ze względu na jego intuicyjność, przejrzystość oraz mnogość dostępnych modułów, które mogą znacznie ułatwić implementację części rozwiązań mających znaleźć zastosowanie w projekcie.

# Wymagania funkcjonalne

**Aplikacja będzie spełniać następujące funkcje:**

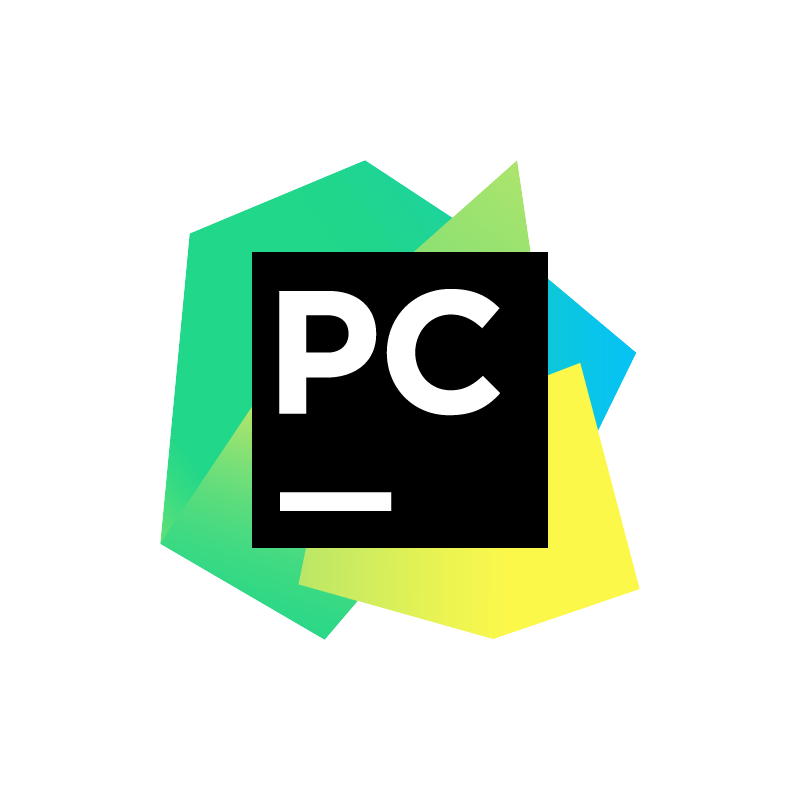
1. Wprowadzanie materiałów do nauki:
   1. Wprowadzanie zagadnień do nauki za pomocą formularza.
   2. Wprowadzenie zagadnień do nauki za pomocą pliku.
2. Wybór języka do nauki popularnych słów.
   1. Wyświetlenie menu z dostępnymi językami:
   2. Generowanie 50 najpopularniejszych słówek w wybranym języku.
3. Nauka wybranego materiału:
   1. Podział materiału na sekcje.
   2. Przetworzenie materiału na zadanie interaktywne i fiszki.
   3. Wygenerowanie funkcji zliczającej punkty.
   4. Generowanie zadań interaktywnych w odpowiednym momencie nauki.
4. Generowanie raportów po zakończeniu sesji.
5. Dostosowanie czasu zegara pomodoro.

# Wymagania niefunkcjonalne

1. Aplikacja powinna działać na komputerach korzystających z systemów Windows 10, Windows 11, Linux oraz MacOS.
2. Interfejs powinien wyświetlać się w trybie okienkowym.
3. Intuicyjna obsługa oraz czytelność.

# Opis wybranego środowiska programistycznego

1. **Pycharm**



Rysunek Logo programu PyCharm

PyCharm to zintegrowane środowisko programistyczne stworzone przez JetBrains   
dla programistów Python. Jest to komercyjny produkt, ale istnieje również darmowa wersja, PyCharm Community Edition, która oferuje wiele podobnych funkcjonalności.

Jedną z kluczowych cech PyCharm jest jego rozbudowane wsparcie dla języka Python, w tym możliwość wskazywania błędów w czasie rzeczywistym i automatyczne uzupełnianie kodu. Oferuje również bogate funkcje debugowania, takie jak interaktywna konsola, wykresy, punkty przerwania i wiele innych.

PyCharm ma również wiele narzędzi do refaktoryzacji kodu, które pomagają programistom utrzymać kod w czystości i łatwości w utrzymaniu. Narzędzia te obejmują automatyczne formatowanie, przesuwanie i zmienianie nazw zmiennych oraz wiele innych.

Inną ważną cechą PyCharm jest integracja z systemami kontroli wersji, takimi jak Git oraz obsługa wielu frameworków Python.

Podczas tworzenia projektu zostanie użytych dużo dodatkowych bibliotek dla języka Python. Najważniejszą z nich będzie biblioteka Tkinter.

Umożliwia ona programistom tworzenie aplikacji z interfejsem graficznym użytkownika (GUI). Tkinter oferuje wiele wbudowanych metod do tworzenia i stylizacji elementów GUI, takich jak przyciski, etykiety, pola tekstowe i wiele innych. Biblioteka ta pozwala również na dodanie interakcji   
do elementów GUI poprzez mechanizmy obsługi zdarzeń.

Inne wykorzystane biblioteki:

* Pandas - umożliwia wczytywanie, przetwarzanie i zapisywanie danych w różnych formatach, takich jak arkusze kalkulacyjne, pliki CSV i wiele innych. Pandas oferuje wiele funkcji   
  do filtrowania, sortowania i grupowania danych, a także możliwość łączenia danych z różnych źródeł.
* Matplotlib – jest używana do tworzenia wykresów i wizualizacji danych. Umożliwia generowanie różnych rodzajów wykresów, takich jak liniowe, słupkowe, punktowe i wiele innych. Matplotlib oferuje również możliwość dostosowania wyglądu tych wykresów.

1. **Sourcery.ai**



Rysunek Logo narzędzia Sourcery.ai

Sourcery.ai to dodatkowy refaktoryzator kodu używający sztucznej inteligencji w celu zwiększenia jakości i przejrzystości pisanego kodu.

1. **Git**



Rysunek Logo narzędzia Git

Rozproszony system kontroli wersji pozwalający na tworzenie, zarządzanie i łączenie zdalnych   
lub lokalnych repozytoriów kodu.

1. Github

**Obraz zawierający czarne, ciemność

Opis wygenerowany automatycznie**

Rysunek 5 Logo serwisu GitHub

GitHub to platforma internetowa zaprojektowana do hostowania i zarządzania projektami   
z wykorzystaniem systemu kontroli wersji Git. Jest popularnym serwisem wykorzystywanym przez programistów, twórców oprogramowania i zespoły do wspólnego tworzenia, udostępniania i zarządzania kodem źródłowym oraz innymi plikami projektu.   
Oto główne cechy i aspekty GitHub-a:

1. System kontroli wersji Git: GitHub opiera się na systemie kontroli wersji Git,   
   który umożliwia śledzenie zmian dokonywanych w kodzie źródłowym projektu. Dzięki temu możliwe jest kontrolowanie historii zmian, cofanie do poprzednich wersji oraz łączenie różnych gałęzi (branchy) kodu.
2. Repozytoria: Wszystkie projekty hostowane na GitHub-ie są przechowywane   
   w repozytoriach. Repozytorium to miejsce, w którym znajdują się wszystkie pliki, foldery, kod źródłowy i inne zasoby projektu.
3. Współpraca: GitHub umożliwia wielu programistom współpracę w jednym projekcie. Użytkownicy mogą zgłaszać błędy (issues), proponować zmiany (pull requests) oraz komentować kod, co ułatwia weryfikację, testowanie i ulepszanie projektu.
4. Branching: GitHub pozwala na tworzenie gałęzi (branchy) kodu, co umożliwia pracę nad różnymi funkcjonalnościami czy rozwiązaniami w izolowanych środowiskach.   
   Po zakończeniu pracy nad gałęzią, można ją połączyć z głównym kodem poprzez tzw. "pull request".

W serwisie github utworzona zostanie tablica kanban w celu lepszego zarządzania zadaniami.

# User stories

Ja jako użytkownik chcę mieć możliwość dodanie do aplikacji materiałów do nauki, aby rozpocząć naukę.

1. Utworzenie możliwości wprowadzania danych do programu poprzez formularz.
2. Utworzenie możliwości wprowadzenia danych do programu za pomocą pliku.

Ja jako użytkownik chcę mieć możliwość zobaczenia raportu po zakończonej sesji nauki, aby posiąść wiedzę odnośnie tego co powinienem powtórzyć.

1. Wygenerowanie raportu na podstawie danych z bieżącej sesji.

Ja jako użytkownik chcę mieć możliwość zresetowania bieżącej sesji nauki, aby rozpocząć naukę od początku.

1. Możliwość zresetowania bieżącej sesji.

Ja jako użytkownik chcę mieć możliwość użycia awaryjnej pauzy podczas sesji nauki, aby zatrzymać chwilowo proces nauki.

1. Możliwość bez konsekwentnego zatrzymania bieżącej sesji za pomocą odpowiedniego przycisku

Ja jako użytkownik chcę mieć możliwość poznania 50 najpopularniejszych słów z danego języka, aby poznać często używane słowa w danym języku.

1. Możliwość nauki najpopularniejszych słów w wybranym języku z menu dostępnego w aplikacji

Ja jako użytkownik chcę mieć możliwość zmiany długości przerw od nauki, aby dostosować proces nauki do swoich wymagań.

1. Możliwość edycji wartości czasu w systemie zliczającym czas nauki.

# Product backlog

1. Utworzenie zdalnego repozytorium
   1. Utworzenie repozytorium
   2. Skonfigurowanie repozytorium
      1. Zainicjowanie repozytorium
      2. Utworzenie pliku gitignore
2. Utworzenie projektu
   1. Zainicjowanie projektu
   2. Skonfigurowanie środowiska programistycznego
3. Zaprojektowanie wyglądu komponentów aplikacji
   1. Zaprojektowanie panelu głównego
   2. Zaprojektowanie panelu wprowadzania materiałów
   3. Zaprojektowanie okna raportów
   4. Zaprojektowanie okna wyboru języka do nauki
   5. Zaprojektowanie wyglądu zegarka pomodoro
   6. Zaprojektowanie gier interaktywnych
      1. Zaprojektowanie wyglądu fiszek
      2. Zaprojektowanie wyglądu gry w wisielca
      3. Zaprojektowanie wyglądu gry prawda/fałsz
      4. Zaprojektowanie wyglądu gry „wskaż poprawne”
      5. Zaprojektowanie gry „dopasuj definicję”
4. Implementacja na podstawie projektu
   1. Zaimplementowanie ekranu głównego
   2. Zaimplementowanie panelu wprowadzania materiałów
      1. Zaimplementowanie wyglądu panelu
      2. Zaimplementowanie funkcji dostępnych z panelu
   3. Zaimplementowanie okna raportu
   4. Zaimplementowanie okna wyboru języka do nauki
      1. Zaimplementowanie wyglądu okna
      2. Zaimplementowanie funkcji dostępnych z poziomu okna
   5. Zaimplementowanie zegarka pomodoro
      1. Zaimplementowanie wyglądu zegarka
      2. Zaimplementowanie funkcjonalności zegarka
      3. Zaimplementowanie możliwości zmiany parametrów zegarka pomodoro
      4. Zaimplementowanie funkcji zatrzymania zegarka pomodoro
5. Implementacja systemów pomocniczych
   1. System oceniający
      1. Zaprojektowanie systemu oceniającego poprawność odpowiedzi
      2. Zaimplementowanie systemu oceniającego poprawność odpowiedzi
   2. System punktowy
      1. Zaprojektowanie systemu zliczającego punkty
      2. Zaimplementowanie systemu zliczającego punkty
   3. System wyświetlający zadania
      1. Zaprojektowanie systemu wyświetlającego zadania
      2. Zaimplementowanie systemu wyświetlającego zadania
6. Implementacja gier interaktywnych
   1. Fiszki
      1. Zaimplementowanie fiszek na podstawie projektu
      2. Zaimplementowanie funkcjonalności fiszek
   2. Gra w wisielca
      1. Zaimplementowanie wyglądu gry na podstawie projektu
      2. Zaimplementowanie funkcjonalności gry w wisielca
   3. Prawda/fałsz
      1. Zaimplementowanie gry prawda/fałsz na podstawie projektu
      2. Zaimplementowanie funkcjonalności gry
   4. „Wskaż poprawne”
      1. Zaimplementowanie gry „wskaż poprawne” na podstawie projektu
      2. Zaimplementowanie funkcjonalności gry
   5. Dopasuj definicję
      1. Zaimplementowanie gry „dopasuj” na podstawie projektu
      2. Zaimplementowanie funkcjonalności gry
7. Zaimplementowanie systemu dobierającego typ gry do danego zadania
   1. Utworzenie schematów dla każdego zadania interaktywnego
   2. Implementacja systemu doboru gry do konkretnego zadania
8. Zaimplementowanie mechanizmu randomizującego zadania
   1. Utworzenie schematu
   2. Implementacja mechanizmu na podstawie schematu