

Autorzy (Informatyka 2023/2024 Semestr 2. - Grupa Ćwiczeniowa nr. 2):

- Krzysztof Hager 52687 - Lider "Kszyszka@Github"
- Elian Godyń 52678 - "Eli3001@Github"
- Jakub Grzęda 52682 - "Keczuk@Github"
- Kacper Grzesik 52684 - "kgrzesik17@Github"
- Michał Fuławka 52675 - "MichalFulawka330@Github"

[Link do repozytorium](https://github.com/Kszyszka/zabka-learning) - <https://github.com/Kszyszka/zabka-learning>

Raport: Projekt Zaliczeniowy - Platforma e-learningowa z quizami i śledzeniem postępów. (*Qmaj Rzabe*)

Wprowadzenie

W tym dokumentcie przedstawiamy proces tworzenia Platformy e-learningowej z quizami i śledzeniem postępów (Web Development) - Qmaj Rzabe (Kumaj Żabe), błąd ortograficzny został popełniony z premedytacją, w celu zwrócenia uwagi.

Założenia

Założenia projektu były następujące:

- Program ma być aplikacją webową (fullstack).
- Program ma korzystać z Frameworku Django, z backendem komunikującym się z bazą sqlite (na skale tego projektu taka baza jest wygodniejszym rozwiązaniem).
- Program ma obsługiwać Panel Użytkownika, z poziomu którego możliwe jest tworzenie własnych testów oraz podglądanie postępów w próbach jego wykonania.
- Program ma obsługiwać wykonywanie testów zarówno jako zalogowany użytkownik jak i Gość.
- Program ma obsługiwać statystyki odpowiedzi do testów, dostępny z poziomu Panelu Użytkownika.

Tworzenie programu

Analizując pomysł na platformę, doszliśmy do następujących konkluzji:

- Projekt będzie podzielony na więcej mniejszych aplikacji, w celu ułatwienia pracy zespołowej i uniknięcia konfliktów przy korzystaniu grupowego z Gita:
 - HomeAPP - Aplikacja odpowiedzialna za stronę domową, jej widoki, logikę oraz przekierowania
 - UsersAPP - Aplikacja odpowiedzialna za zarządzanie użytkownikami, zbudowana na podstawie Django Auth, pozwala m.in. na logowanie i rejestrację.
 - QuizAPP - Aplikacja służąca do obsługi Panelu Użytkownika, oraz wszelkich działań związanych z zarządzaniem testami z jego poziomu (główny moduł programu, przeznaczony dla

zalogowanych użytkowników).

- TestsAPP - Aplikacja służąca do obsługi wykonywania testów zarówno przez zalogowanych użytkowników jak i Gości.

Projekt pozwolił członkom zespołu pierwszy raz korzystać z gita w praktyczny sposób przy kolaboracyjnej pracy. Dzięki jego wykorzystaniu możliwe były powroty do działających wersji oraz szybka współpraca przy tworzeniu zależnych od siebie modułów.

Wybór Frameworka Django do wykonania projektu okazał się dobrym pomysłem. Dzięki niemu byliśmy w stanie w nowoczesny sposób podzielić się pracą na każdą aplikację, współpracować oraz konsultować się co do funkcjonalności i ją modyfikować, bez zepsucia ogółu programu.

Zespół podzielił się wstępnie na front-end:

- Jakub Grzęda
- Michał Fuławka

Oraz back-end:

- Krzysztof Hager
- Eliań Godyń
- Kacper Grzesik

Jednak mimo to, każdy członek zespołu miał styczność zarówno z front-endem (HTML, CSS, JS) jak i back-endem (Python). Dzięki temu wszystkie aplikacje mogły być równolegle rozwijane z obu stron.

Podjęliśmy technikę mergowania branchy na repozytorium, w celu zachowania informacji o wszystkich commitach oraz ich historii (dostępne na Github, wraz z przypisanymi osobami). Backlog projektu był prowadzony w GitHub Projects - <https://github.com/users/Kszyszka/projects/2>

Napotkane problemy

1. Brak znajomości Frameworka Django - skok na głęboką wodę.
2. Problemy ze zrozumieniem logiki, views, struktury projektu narzuconej przez Django (static, templates, apps, etc.).
3. Rozszerzenie Django Auth.
4. Zrozumienie Jinja2
5. Problemy z wieloosobową obsługą Git.
6. Problemy przy zaawansowanej obsłudze POST.
7. Problemy z integracją Pythona z HTML oraz automatycznym generowaniem zawartości stron.

Rozwiązanie problemów

1. Konsultacje z członkami zespołu, tutoriale, wspomaganie się dokumentacją Django.
2. Konsultacje z członkami zespołu, metoda prób i błędów do momentu samodzielnego zrozumienia.
3. Zmodyfikowanie wykorzystania Auth w innych aplikacjach, w celu lepszej współpracy z domyślnym Django Auth.
4. Konsultacje z członkami zespołu, metoda prób i błędów do momentu samodzielnego zrozumienia.
5. Nauka nt. Gita z oficjalnej dokumentacji, zgłębienie informacji m.in. o Merge Conflictach, dzielenie się wiedzą w zespole.

6. Przepuszczenie przez JavaScript jako middleware.
7. Poszerzenie wiedzy z zakresu Jinja2.

Podsumowanie

Wykonanie projektu pozwoliło każdemu członkowi na praktyczne zapoznanie się z Gitem, Pythonem przy Web Developmencie oraz Frameworkiem Django. Początkowo skok na głęboką wodę poprzez wybranie nieznanych przez nikogo z zespołu technologie był problematyczny, spowolnił pracę i tworzył problemy w morale zespołu. Jednak po czasie, kiedy każdy zdążył się już zapoznać z wymaganymi technologiami, wybór Django okazał się dobrym pomysłem.

Pozwoliło to nam stworzyć prototyp aplikacji zgodnie z nowoczesnymi standardami oraz uczestniczyć w projekcie w sposób podobny jak byłoby to wykonane w środowisku komercyjnym.

Wykorzystane technologie

- Python
- Django
- Bootstrap
- JavaScript
- CSS
- SQLite
- Git

Źródła wiedzy

- <https://docs.djangoproject.com/en/5.0/>
- <https://jinja.palletsprojects.com/en/3.1.x/>
- <https://stackoverflow.com/>
- <https://www.youtube.com/@TechWithTim>
- <https://www.youtube.com/@crycetruly>
- <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
- <https://www.youtube.com/@code-4-design>
- ChatGPT