

INTERNETOWA PLATFORMA EDUKACYJNA DO KURSÓW PROGRAMISTYCZNYCH

Oskar Bienko, Katarzyna Kazimierska-Drobny, Piotr Prokopowicz

Uniwersytet Kazimierza Wielkiego
Wydział Informatyki, Wydział Mechatroniki
ul. Mikołaja Kopernika 1, 85-074 Bydgoszcz
e-mail: oskar.bienko@student.ukw.edu.pl

Streszczenie: Niniejsza publikacja opisuje proces analizy projektowej oraz implementacji aplikacji webowej opartej na frameworku Django, zaprojektowanej w celu umożliwienia zdobywania umiejętności programowania za pomocą kursów online. Aplikacja ta pozwala użytkownikom na tworzenie i udostępnianie kursów, w tym zarządzanie lekcjami, quizami, zasobami edukacyjnymi oraz systemem ocen i komentarzy. Platforma oferuje również funkcjonalności związane z zarządzaniem zamówieniami i płatnościami, co umożliwia pełną obsługę kursów online. Dodatkowo, szczegółowo opisano zastosowane technologie oraz funkcje, które oferuje system.

Słowa kluczowe: platforma edukacyjna, platforma e-learningowa, e-learning, kursy online, kursy programistyczne, nauczanie zdalne, Python, Django, aplikacja webowa

ONLINE LEARNING PLATFORM FOR PROGRAMMING COURSES

Abstract: This article describes the design analysis and implementation of a web application based on the Django framework, designed to facilitate learning programming skills through online courses. The application allows users to create and share courses, including managing lessons, quizzes, educational resources, and a system for ratings and comments. The platform also offers functionalities related to order and payment management, providing a complete solution for online course handling. Additionally, the technologies used and the features offered by the system are thoroughly described.

Keywords: educational platform, e-learning platform, e-learning, online courses, programming courses, distance learning, Python, Django, web application

1. Wstęp

Rynek IT stanowi dynamicznie rozwijającą się przestrzeń, która nieustannie ustala nowe trendy i innowacje. Proces przyswajania nowości technologicznych może być czasochłonny, co sprawia, że wielu programistów boryka się z brakami w umiejętnościach, które są niezbędne do efektywnej pracy w danej dziedzinie. Zgodnie z raportem [1] z 2023 roku opracowanym przez portal pracy NoFluffJobs, aż 52,8% kandydatów aplikowało na oferty pracy, spełniając jedynie połowę wymagań, a 48,4% aplikowało, nie znając dobrze wymaganej technologii. Luki te stanowią

barierę dla rozwoju zawodowego, co czyni wejście na rynek pracy, szczególnie dla osób rozpoczynających swoją karierę w branży IT, dużym wyzwaniem.

Stałe zmiany technologiczne i nowe wymagania w branży IT wymuszają na specjalistach nieustanne poszerzanie wiedzy. W efekcie rosną zarówno wymagania związane z procesem uczenia się, jak i te stawiane przez rynek pracy. Każdy programista musi być gotowy do ciągłego dostosowywania się do nowych realiów. Z kolei konkurencyjny rynek pracy oczekuje od pracowników inicjatywy i

kreatywności. Przystosowanie się do tych zmian wymaga czasu i zaangażowania w proces uzupełniania braków, co stanowi nie tylko szansę, ale również konieczność na utrzymanie zatrudnienia, czy podjęcie satysfakcjonującej pracy [2].

Celem pracy jest zaprojektowanie i implementacja internetowej platformy edukacyjnej, która umożliwia zdobywanie umiejętności programowania poprzez kursy online. Aplikacja ma na celu wypełnianie luk kwalifikacyjnych wśród programistów poprzez umożliwienie im dostępu do kursów tworzonych przez profesjonalistów, którzy chcą dzielić się swoją wiedzą. Jest to przestrzeń wymiany doświadczeń, która stanowi realną szansę na zdobycie pożądaných na rynku kompetencji.

2. Projekt internetowej platformy edukacyjnej

Proces projektowania i implementacji internetowej platformy edukacyjnej oparto na sprawdzonych metodach. Rozpoczęto od określenia założeń systemu, jego ograniczeń oraz poszczególnych ról i scenariuszy. Stworzono diagram przypadków użycia oraz diagram klas, które pozwoliły zidentyfikować kluczowe funkcjonalności systemu oraz relacje pomiędzy poszczególnymi komponentami. Wykorzystano różnorodne technologie dla backendu, frontendu, bazy danych oraz integracji zewnętrznych systemów, aby zapewnić aplikacji funkcjonalność, wydajność oraz łatwość użytkowania [3].

2.1 Analiza projektowa

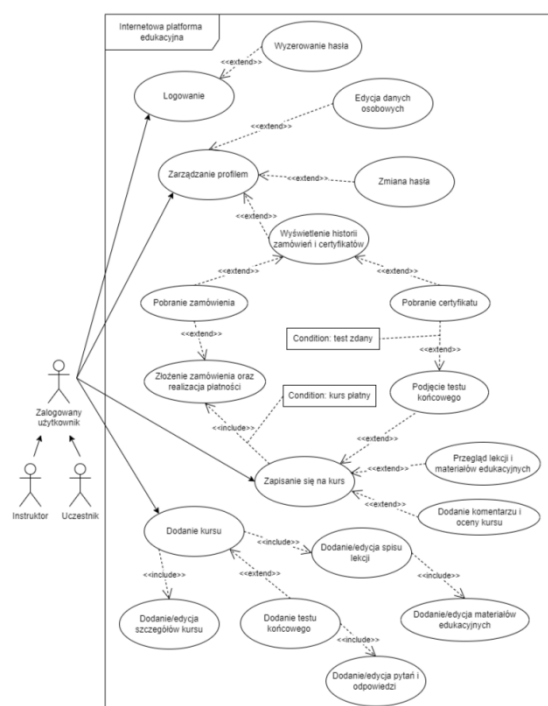
W celu zidentyfikowania głównych funkcji aplikacji oraz interakcji użytkowników, opracowano diagram przypadków użycia. Ujęto w nim trzy role użytkowników:

- Gość - może przeglądać kursy i rejestrować się w systemie.
- Zalogowany użytkownik - podzielona jest na rolę instruktora oraz uczestnika, może zapisywać się na kursy, przeglądać materiały, uczestniczyć w quizach oraz zarządzać swoimi kursami.
- Administrator - posiada pełne prawa do zarządzania użytkownikami, kursami, kategoriami, zamówieniami itd.

Każda z ról została opisana szczegółowymi scenariuszami, w tym m.in. procesem rejestracji, tworzeniem kursów czy dodawaniem materiałów dydaktycznych. Przykładowo, w przypadku zalogowanego użytkownika uwzględniono takie działania jak:

- przeglądanie i zapisywanie się na kursy,
- tworzenie własnego kursu oraz quizu,
- uczestnictwo w lekcjach i quizach,
- dodawanie opinii i ocen kursów,
- zarządzanie profilem oraz historią zamówień.

Diagram przypadków użycia dla zalogowanego użytkownika (Rys. 1) ilustruje te kluczowe interakcje, pokazując, w jaki sposób użytkownik komunikuje się z różnymi funkcjami systemu. Ten diagram stał się podstawą do identyfikacji wymagań funkcjonalnych oraz szczegółowego planowania implementacji poszczególnych modułów.

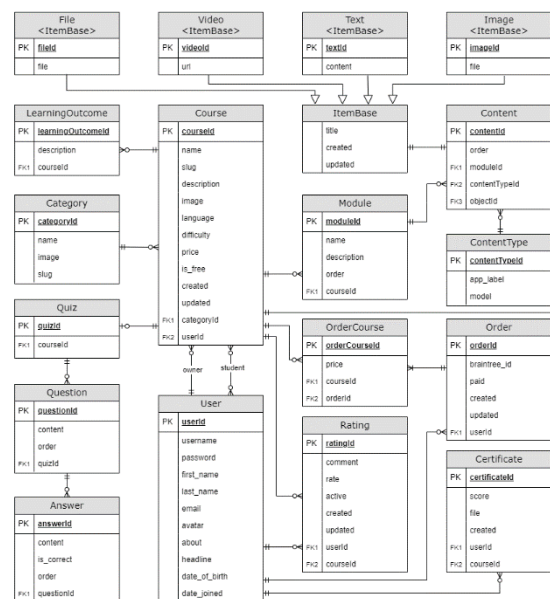


Rys. 1. Diagram przypadków użycia aktora Zalogowany użytkownik.

Na bazie analizy przypadków użycia opracowano diagram związków encji, który przedstawia graficzną reprezentację relacji pomiędzy obiektami w systemie informatycznym. Stanowi wizualny punkt wyjścia do projektowania bazy danych. Diagram ten zawiera różne symbole i złącza, które wizualizują dwie ważne informacje: główne podmioty w zakresie systemu oraz

wzajemne powiązania między tymi podmiotami. Takie podmioty nazywane są encjami, zwykle zawierają również atrybuty, czyli właściwości używane do jej opisu [4].

Diagram (Rys. 2) na szczycie hierarchii uwzględnia między innymi użytkownika czy kategorię. To oznacza, że do stworzenia obiektu nie wymaga się uczestnictwa innych encji. Kategorie, zakłada się, że są stałe, zdefiniowane wcześniej jako języki programowania: C#, CSS, HTML, Java, JavaScript, PHP, Python, SQL. Pozwalają na uporządkowanie kursów, definiują ich tematykę oraz ułatwiają początkową nawigację. Głównym elementem są same kursy. Konkretnie wystąpienie tej encji może wystąpić wyłącznie w kontekście konkretnego użytkownika (autora) i kategorii. Natomiast przedstawiona relacja wiele do wielu odnosi się do uczestników kursu, gdzie użytkownik może zapisać się na wiele kursów, a kurs może posiadać wiele uczestników. Opinie zostawione przez uczestników zdefiniowane są encją „Rating” oraz zawiera między innymi komentarz i ocenę w skali od 1 do 5. Do kursu można dodać wiele efektów kształcenia i modułów (lekcji). Kolejnym ważnym elementem jest system quizów. Kurs może być powiązany z jednym quizem. Dalej testy składają się z wielu pytań, a te kolejno z wielu odpowiedzi. Obie encje zawierają atrybut „order”, aby umożliwić autorowi manipulację kolejnością wyświetlanych dla uczestników treści. Certyfikat natomiast, definiowany jest kluczem obcym kursu i uczestnika, uzyskanym wynikiem z quizu, datą oraz jest dostępny jako plik w formacie PDF. Istnieje system obsługujący zamówienia, w którego skład wchodzi encja „Order” oraz ta pozwalająca na powiązanie zamówienia z kursem. Do każdego zamówienia należy przypisać identyfikator transakcji zewnętrznego systemu. Umożliwi to powiązanie każdego zamówienia z odpowiednią transakcją, informując, czy zostało opłacone. Ze względu na wykorzystywany framework oraz użyte moduły, poza wymienionymi encjami istnieje szereg dodatkowych, których nie zamieszczono na diagramie, ale są równie istotne. Dotyczą one uprawnień, grup użytkowników, integracji z mediami społecznościowymi, witryn, sesji użytkownika czy miniatur obrazów. Ich rola polega na dostarczeniu możliwości odpowiedniej administracji systemem i umożliwia istnienie reszty usług potrzebnych do funkcjonowania platformy.



Rys. 2. Diagram związków encji.

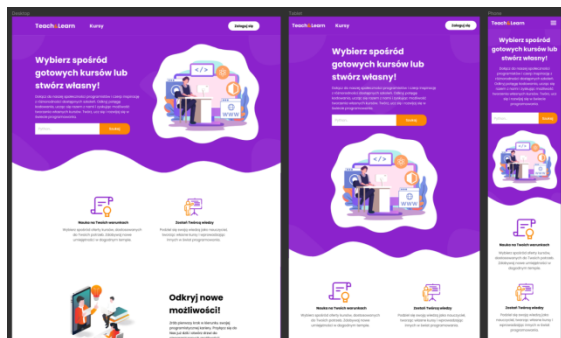
Diagram umożliwił uporządkowanie modelu danych oraz ułatwił zastosowanie podejścia DRY (Don't Repeat Yourself), co pozwoliło na optymalizację kodu.

Na podstawie dotychczasowej analizy oraz zaprojektowanego diagramu związków encji, stworzono schemat relacyjnej bazy danych w odniesieniu do PostgreSQL. Poniżej (Rys. 3) przedstawiono wycinek schematu dla internetowej platformy edukacyjnej, pokazano zaledwie część z istniejących tabel w aplikacji. Konwersja diagramu ERD w schemat relacyjnej bazy danych polegała na przekształceniu encji i atrybutów w konkretne tabele i kolumny. Nazwy tabel otrzymały przedrostek w postaci nazwy aplikacji, w której zostały zaimplementowane. Nie zostały utworzone tabele dla encji abstrakcyjnych. Każdemu atrybutowi przypisywany jest odpowiedni typ oraz wprowadzane są ograniczenia takie jak unikalność czy określenie możliwości zostawienia pustego pola. Relacje reprezentowane są przez klucze główne oraz obce. Dodatkowo, relacja wiele do wielu przekształca się w dodatkową tabelę, nazywaną haszującą.

public.accounts_user_groups id integer NOT NULL user_id integer NOT NULL group_id integer NOT NULL	public.accounts_user id integer NOT NULL password character varying(128) NOT NULL last_login timestamp with time zone is_superuser boolean NOT NULL username character varying(150) NOT NULL first_name character varying(30) NOT NULL last_name character varying(150) NOT NULL email character varying(254) NOT NULL is_staff boolean NOT NULL is_active boolean NOT NULL date_joined timestamp with time zone NOT NULL avatar character varying(100) NOT NULL data_of_birth date about text headline character varying(64)	public.courses_course id integer NOT NULL name character varying(70) NOT NULL slug character varying(200) NOT NULL description text image character varying(100) NOT NULL language character varying(3) NOT NULL price numeric(15,2) NOT NULL created timestamp with time zone NOT NULL updated timestamp with time zone NOT NULL category_id integer NOT NULL owner_id integer NOT NULL is_free boolean NOT NULL difficulty character varying(20) NOT NULL
public.accounts_user_permissions id integer NOT NULL user_id integer NOT NULL permission_id integer NOT NULL	public.courses_course_students id integer NOT NULL course_id integer NOT NULL user_id integer NOT NULL	public.courses_rating id integer NOT NULL comment text rate integer NOT NULL created timestamp with time zone NOT NULL updated timestamp with time zone NOT NULL active boolean NOT NULL course_id integer NOT NULL user_id integer NOT NULL
public.auth_group id integer NOT NULL name character varying(150) NOT NULL	public.django_admin_log id integer NOT NULL action_time timestamp with time zone NOT NULL object_id text object_repr character varying(200) NOT NULL action_flag smallint NOT NULL change_message text NOT NULL content_type_id integer user_id integer NOT NULL CONSTRAINT django_admin_log_action_flag_check	public.orders_order id integer NOT NULL created timestamp with time zone NOT NULL updated timestamp with time zone NOT NULL paid boolean NOT NULL braintree_id character varying(150) NOT NULL user_id integer NOT NULL
public.auth_group_permissions id integer NOT NULL group_id integer NOT NULL permission_id integer NOT NULL	public.django_content_type id integer NOT NULL app_label character varying(100) NOT NULL model character varying(100) NOT NULL	public.orders_ordercourse id integer NOT NULL price numeric(15,2) NOT NULL course_id integer NOT NULL order_id integer NOT NULL
public.auth_permission id integer NOT NULL name character varying(255) NOT NULL content_type_id integer NOT NULL codename character varying(100) NOT NULL	public.django_migrations id integer NOT NULL app character varying(255) NOT NULL name character varying(255) NOT NULL applied timestamp with time zone NOT NULL	public.social_auth_association id integer NOT NULL server_url character varying(255) NOT NULL handle character varying(255) NOT NULL secret character varying(255) NOT NULL issued integer NOT NULL lifetime integer NOT NULL extra_type character varying(64) NOT NULL
public.django_session id integer NOT NULL session_data text NOT NULL expire_date timestamp with time zone NOT NULL	public.django_site id integer NOT NULL domain character varying(100) NOT NULL name character varying(50) NOT NULL	
public.courses_category id integer NOT NULL name character varying(200) NOT NULL slug character varying(200) NOT NULL image character varying(200)	public.static_precompiler_dependency id integer NOT NULL source character varying(255) NOT NULL depends_on character varying(255) NOT NULL	
public.courses_learningoutcome id integer NOT NULL description text NOT NULL course_id integer NOT NULL	public.easy_thumbnails_source id integer NOT NULL storage_path character varying(40) NOT NULL name character varying(255) NOT NULL modified timestamp with time zone NOT NULL	

Rys. 3. Wycinek schematu relacyjnej bazy danych.

Interfejs użytkownika zaprojektowano z wykorzystaniem narzędzia Figma, co pozwoliło na stworzenie spójnego i responsywnego wyglądu aplikacji. W celu uzyskania aplikacji poprawnie wyświetlającej się na różnych typach urządzeń, starannie zaprojektowano wygląd dla większości stron w aplikacji. Dla każdej z nich indywidualnie opracowano projekt, uwzględniając różnice w rozmiarach (przykładowe rozmieszczenie pokazano na Rys. 4) [5].



Rys. 4. Wycinek widoku strony głównej z poziomu niezalogowanego użytkownika, wyświetlany na trzech typach urządzeń.

Szczególny nacisk położony został na estetykę oraz atrakcyjność interfejsu użytkownika pod względem wizualnym. Zdecydowano się na konkretną paletę kolorów, zastosowano spójne style, co pomaga w utrzymaniu jednolitego tonu. Wspólna czcionka na wszystkich stronach aplikacji nie wprowadza chaosu i nie utrudnia przeszukiwania informacji. To wszystko wpływa na pozytywne wrażenia osoby korzystającej i przekłada się na budowanie zaufania do aplikacji.

2.2 Podejście technologiczne

Proces rozwoju opierał się na modelu kaskadowym, zwanym również waterfall, co umożliwiło etapowe realizowanie zadań: od analizy wymagań, przez projektowanie i implementację, po testy i wdrożenie. Dzięki zastosowaniu metodologii SMART cele projektu zostały precyzyjnie określone i skutecznie osiągnięte.

Językiem programowania wybranym do implementacji backendu był Python, znany ze swojej prostoty, czytelności oraz wsparcia szerokiej społeczności programistów [6, 7]. Projektowana aplikacja oparta jest na frameworku Django, który zajmuje się większością problemów związanych z tworzeniem stron internetowych, jest darmowy i open source [8, 9]. Framework ten zapewnia wydajność, bezpieczeństwo oraz możliwość łatwego skalowania, co było kluczowym aspektem przy wyborze technologii do realizacji założonych celów [10]. Posiada wbudowane zabezpieczenia przed atakami takimi jak SQL injection, XSS czy CSRF [11]. Ze względu na możliwość szybkiego tworzenia aplikacji webowych w oparciu o architekturę MVT (Model-View-Template), która jest szczególnym przypadkiem MVC, doskonale sprawdził się jako fundament systemu [12].

Dodatkowo, w aplikacji zaimplementowano szereg integracji i narzędzi wspierających. Wysyłka e-maili została zrealizowana za pomocą usługi SendGrid, która umożliwiła wysyłanie informacji o resetowaniu hasła. Obsługa płatności została wdrożona z wykorzystaniem bramki Braintree, co zapewniło bezpieczne transakcje online i wsparcie dla różnych metod płatności. Generowanie certyfikatów ukończenia kursów w formacie PDF umożliwiła biblioteka WeasyPrint. Dodatkowo zastosowano moduł DjangoSocialAuth, co umożliwiło wdrożenie funkcji logowania z wykorzystaniem mediów społecznościowych, takich jak Google czy Facebook [13].

Struktura interfejsu użytkownika powstała w oparciu o HTML5 i CSS3, przy czym zastosowano preprocesor Less, który umożliwił bardziej zaawansowaną organizację stylów dzięki zmiennym i zagnieżdżeniom. W celu zapewnienia dynamicznej interakcji z użytkownikiem wykorzystano język JavaScript oraz technologię AJAX, by dać instruktorowi zarządzającemu swoimi kursami większe możliwości. Wprowadzono funkcję umożliwiającą operację przeciągania i upuszczania modułów kursów oraz ich zawartości, w celu odpowiedniego uszeregowania. Interaktywna modyfikacja zmiany kolejności odbywa się bez potrzeby odświeżania strony, co pozytywnie wpływa na doświadczenia użytkownika.

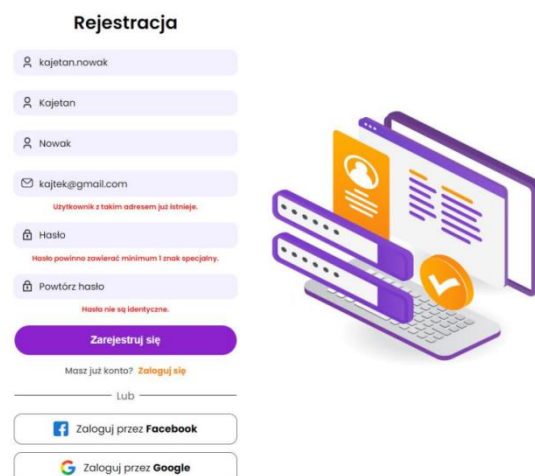
Platforma bazowała na systemie zarządzania relacyjną bazą danych PostgreSQL, który został wybrany ze względu na swoją skalowalność i wsparcie zaawansowanych funkcji, takich jak pełnotekstowe wyszukiwanie. W projekcie użyto wyszukiwanie z podobieństwem trygramu, ponieważ metoda ta jest skuteczna w znajdowaniu słów, które są podobne pod względem literówek, co jest ważne w przypadku, gdy użytkownicy mogą nazywać kursy według upodobania. Przykładowo, mimo, że użytkownik wyszukuje „aplikacje webowe”, w rezultacie otrzymał kursy zawierające frazy „aplikację web” oraz „aplikacji webowych”. Cała metoda polega na zliczaniu w dwóch wyrazach, wspólnych trygramów, czyli grup kolejno trzech takich samych znaków. W tym celu użyto rozszerzenia pg_trm, które udostępnia funkcje i operatory służące do określania podobieństwa tekstu alfanumerycznego na podstawie dopasowania trygramowego, a także klasy operatorów indeksujących, które wspierają szybkie wyszukiwanie podobnych ciągów [14].

3. Testowanie internetowej platformy edukacyjnej

W wyniku realizacji projektu stworzono w pełni funkcjonalną platformę edukacyjną, która umożliwia efektywne zarządzanie kursami online oraz ich udostępnianie użytkownikom. Aplikacja spełnia założone cele projektowe i oferuje kluczowe funkcjonalności, które przyczyniają się do wspierania nauczania zdalnego.

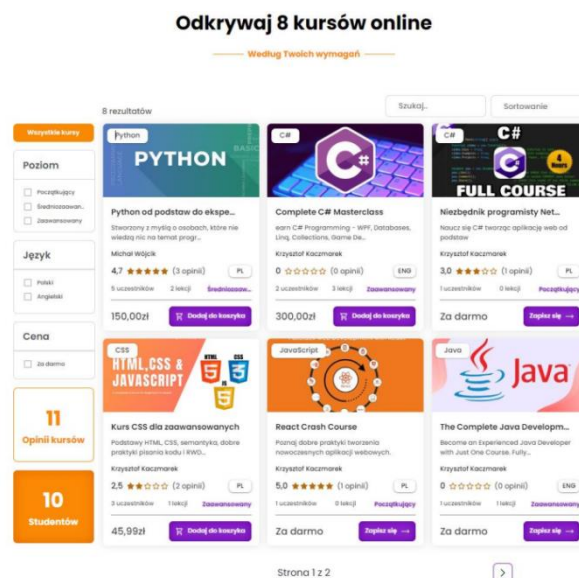
Projekt aplikacji e-learningowej wyróżnia się estetycznym, przejrzystym i funkcjonalnym interfejsem użytkownika, zaprojektowanym z myślą o intuicyjnej nawigacji i łatwości korzystania

zarówno dla nowych, jak i zaawansowanych użytkowników.



Rys. 5. Zrzut ekranu strony rejestracji.

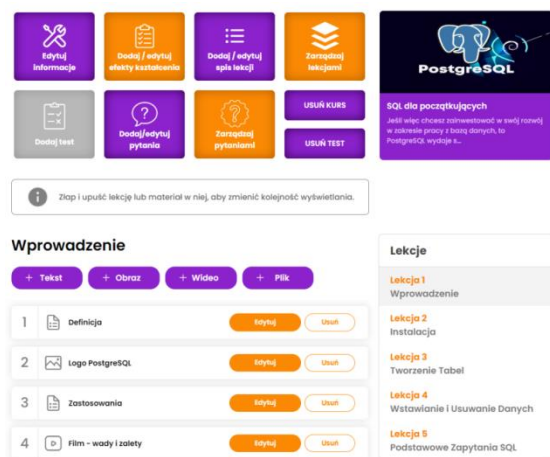
Zrzut ekranu (Rys. 5) ukazuje proces rejestracji nowego użytkownika. Formularz rejestracyjny jest podzielony na intuicyjne pola, wymagające wprowadzenia danych takich jak imię, nazwisko, adres e-mail oraz hasło. Mechanizmy walidacji danych w czasie rzeczywistym informują użytkownika o wymaganiach dotyczących haseł czy błędach w formularzu (np. podwójne wpisanie hasła). Dodatkowo użytkownik może skorzystać z opcji rejestracji za pośrednictwem kont Google lub Facebook, co znacząco ułatwia proces tworzenia konta.



Rys. 6. Zrzut ekranu części strony z listą kursów.

Na załączonym zrzucie ekranu (Rys. 6) przedstawiono stronę główną aplikacji, prezentującą listę dostępnych kursów online. Widok ten zawiera kafelki kursów, gdzie każdy kafelek

przedstawia kluczowe informacje, takie jak nazwa kursu, język prowadzenia, poziom trudności, liczba opinii użytkowników oraz cena (lub informacja o darmowym dostępie). Filtry po lewej stronie pozwalają użytkownikowi dostosować wyniki wyszukiwania według poziomu trudności, języka kursu oraz ceny. Dodatkowe opcje, takie jak wyszukiwarka i możliwość sortowania wyników, zwiększają funkcjonalność tej sekcji. Dzięki takiemu podejściu, użytkownik może szybko i łatwo odnaleźć kursy dopasowane do swoich potrzeb.



Rys. 7. Zrzut ekranu części panelu instruktora.

Na zrzucie ekranu (Rys. 7) zaprezentowano jeden z widoków dla instruktora, który pozwala zarządzać treścią kursu. Użytkownik ma dostęp do funkcji takich jak edytowanie informacji o kursie, dodawanie efektów kształcenia, listy lekcji czy pytań testowych. W dolnej części ekranu znajduje się lista lekcji wchodzących w skład kursu, z możliwością dodawania materiałów w różnych formatach (tekst, obraz, wideo, plik). Struktura ta umożliwia dynamiczne dostosowywanie zawartości kursu do zmieniających się potrzeb uczniów.

Aplikacja oferuje szeroką gamę dodatkowych widoków, które wspierają różnorodne funkcjonalności i podnoszą komfort użytkownika. Oprócz omówionych już sekcji, użytkownicy mają dostęp do widoków takich jak logowanie oraz przywracanie hasła, które zostały zaprojektowane z myślą o bezpieczeństwie i wygodzie. Szczegóły kursu prezentują pełne informacje na temat wybranego szkolenia, w tym jego strukturę, opisy, recenzje oraz możliwość zapisania się lub zakupu. Dla instruktorów i uczestników przewidziano także zaawansowane funkcje związane z quizami, w tym widok tworzenia quizów, który pozwala na definiowanie pytań, odpowiedzi oraz ich kolejności, a także widok rozwiązywania quizów, w którym użytkownik może sprawdzić swoją wiedzę.

Dodatkowo aplikacja umożliwia przeglądanie historii zamówień, co pozwala użytkownikom śledzić zakupione kursy i transakcje, oraz oferuje dostęp do certyfikatów, które mogą być pobierane i wykorzystywane jako dowód zdobytych umiejętności. Poza tym w aplikacji znajduje się wiele innych widoków wspierających różnorodne procesy, co czyni ją kompleksowym narzędziem do nauki i zarządzania edukacją online.

W aplikacji wykonano zarówno testy manualne jak i automatyczne. Przetestowano formularze, wprowadzając różnorodne scenariusze. Sprawdzone, czy formularze odpowiednio wyświetlają komunikaty błędów oraz ich lokalizację. Zweryfikowano, jak aplikacja reaguje na skrajne przypadki, takie jak puste pola lub bardzo długie ciągi znaków. Przeprowadzono testy dynamicznych elementów na stronie, takich jak listy rozwijane czy przyciski. Zweryfikowano poprawne działanie funkcji wyszukiwających, sortujących i stronicujących. Upewniono się, że aplikacja wyświetla się i działa prawidłowo na różnych typach urządzeń, czy jest responsywna. Sprawdzone, czy użytkownik bez odpowiednich uprawnień nie ma dostępu do zależnych od tego zasobów i stron.

Napisano automatyczne testy jednostkowe, które koncentrują się na sprawdzeniu poszczególnych fragmentów kodu. Izolują odpowiednią sekcję kodu i weryfikują jej poprawność. Testy jednostkowe mogą sprawdzać pojedyncze funkcje, metody, moduły czy obiekty. W platformie edukacyjnej sprawdzają poprawność:

- tworzenia kursów oraz ich właściwości, efektów kształcenia, modułów, konkretnych,
- typów treści, dostępu do kursów i zasobów,
- przekierowań do odpowiednich widoków
- dodawania i usuwania kursów z koszyka, wyświetlania zawartości koszyka,
- tworzenia zamówienia,
- procesu płatności udanej i nieudanej,
- procesu rejestracji użytkownika i przypisania do grup,
- wyświetlenia błędów formularzy,
- tworzenia quizów i dostępu do nich.

Wszystkie testy zostały przeprowadzone pomyślnie, co potwierdza poprawność działania aplikacji i gotowość do użytkowania.

4. Podsumowanie i wnioski

Opracowana aplikacja edukacyjna, wspierająca procesy nauczania online, jest znaczącym krokiem w kierunku wykorzystania nowoczesnych technologii do dostarczania wiedzy w sposób bardziej interaktywny i dostępny. Wyniki projektu wskazują, że zastosowanie zaawansowanych narzędzi technologicznych, takich jak Django, pozwala na budowanie skalowalnych, elastycznych i funkcjonalnych rozwiązań, które spełniają zarówno wymagania użytkowników, jak i współczesne standardy projektowania aplikacji webowych.

Stworzona aplikacja webowa posiada szereg funkcjonalności zakładanych na etapie projektowym, które pozwalają na dodawanie oraz uczestniczenie w kursach online. Dzięki temu, możliwe jest uczenie się wcześniej z dodanych kursów przez osoby bardziej doświadczone, chcące dzielić się swoją wiedzą. Oznacza to, że użytkownik decydujący się zapisać lub zakupić kurs, dostaje dostęp do treści, niezależnie od miejsca i czasu. Pomaga to w wypełnianiu luk kwalifikacyjnych programistów, dostarczając odpowiednich narzędzi do nauki, jednocześnie nie ingerując w indywidualne ich harmonogramy dnia.

Podczas realizacji projektu napotkano kilka wyzwań, które ograniczyły pełne wykorzystanie potencjału aplikacji. Przykładem jest ograniczona liczba testów realnych użytkowników, co mogło wpłynąć na ostateczną ocenę intuicyjności interfejsu. Dodatkowo, zależność od zewnętrznych dostawców usług, takich jak SendGrid czy Braintree, chociaż korzystna z punktu widzenia szybkości wdrożenia, może stwarzać problemy związane z potencjalnymi zmianami w polityce tych usług.

Platforma edukacyjna posiada liczne funkcjonalności, które już na ten moment umożliwiają efektywne użytkowanie. Jednakże istnieją możliwości rozwoju, które wpłynęłyby na pozytywne doświadczenia użytkownika oraz atrakcyjność aplikacji. Rozszerzenie systemu o czat umożliwiający komunikację pomiędzy trenerem a uczestnikiem kursu stanowi interesujący punkt rozwoju. Czat ten może służyć do natychmiastowej wymiany informacji, pytań czy sugestii dotyczących materiałów edukacyjnych. Umożliwiłoby to szybkie reagowanie na problemy osób zapisanych oraz odpowiedzi na pytania i bieżące wsparcie. Dodatkowym obszarem do rozwinięcia jest implementacja mechanizmów gamifikacji, które mogłyby podnieść motywację

użytkowników do korzystania z aplikacji. Jako perspektywa rozwoju, istnieje możliwość dodania interaktywnych ćwiczeń i zadań do lekcji. Wymagałyby one między innymi pisanie fragmentów kodu, czyli większego zaangażowania i wkładu. Takie praktyczne zastosowanie zdobytej wiedzy podniosłoby wartość platformy. Użytkownicy mogliby doskonalić programowanie w kontrolowanym środowisku i natychmiastowo otrzymywać wyniki dotyczące rozwiązania.

Podsumowując, opracowany system stanowi solidną podstawę do dalszego rozwoju, a uzyskane wyniki potwierdzają jego potencjał jako narzędzia wspierającego edukację online. Jednocześnie istnieje wiele możliwości rozbudowy i optymalizacji, co otwiera drogę do dalszych prac badawczo-rozwojowych, które mogą jeszcze bardziej zwiększyć użyteczność i wszechstronność aplikacji.

Literatura

- [1] https://wp-insights.s3.eu-central-1.amazonaws.com/wp-content/uploads/2023/10/26093840/Tak-mnie-rekrutuj_Candidate-Experience-2023-No-Fluff-Jobs.pdf
- [2] Chabior, A. (2018). Edukacja i kształcenie ustawiczne – elementy diagnozy. *Edukacja Ustawiczna Dorosłych*, 4(103).
- [3] Alhumaidan, Fahad. (2012). A Critical Analysis and Treatment of Important UML Diagrams Enhancing Modeling Power. *Intelligent Information Management*. 04. 10.4236/iim.2012.45034.
- [4] Song, I. Y., Evans, M., & Park, E. K. (1995). A comparative analysis of entityrelationship diagrams. *Journal of Computer and Software Engineering*, 3(4), 427-459.
- [5] Interaction Design Foundation - IxDF. 2022. *Responsive Design: Best Practices*
- [6] Van Rossum, G. (2003). An introduction to Python (p. 115). F. L. Drake (Ed.). Bristol: Network Theory Ltd..
- [7] Lutz, M. (2010). *Programming python*. "O'Reilly Media, Inc."
- [8] Forcier, J., Bissex, P., & Chun, W. J. (2008). *Python web development with Django*. Addison-Wesley Professional.
- [9] Vincent, W. S. (2022). *Django for Beginners: Build websites with Python and Django*. WelcomeToCode.
- [10] Thakur, P., & Jadon, P. (2023, May). Django: Developing web using Python. In *2023 3rd*