



Wydział Finansów i Bankowości
(Times New Roman 12 punktów)

Dziennik Elektroniczny

PROJEKT DYPLOMOWY

Poznań 2024

DANE PARTNERÓW

A1. Dane Promotora [Podpisy są wymagane wyłącznie, gdy składana jest papierowa wersja projektu.

Czcionka Times New Roman, 12pkt]

Imię i nazwisko	
Stopień / Tytuł naukowy	
Data i podpis	

A2. Dane członków Zespołu projektu [Czcionka Times New Roman, 12pkt]

Imię i nazwisko	Maciej Szewczuk
Kierunek studiów	Informatyka
Tryb studiów	Stacjonarne
Data i podpis	

Imię i nazwisko	Adam Stankiewicz
Kierunek studiów	Informatyka
Tryb studiów	Stacjonarne
Podpis	

Imię i nazwisko	Jakub Kowalski
Kierunek studiów	Informatyka
Tryb studiów	Stacjonarne
Data i podpis	

ZAŁOŻENIA PROJEKTU

B1. Opis projektu

Uzasadnienie wyboru tematu

Współczesne systemy edukacyjne stoją przed wyzwaniem skutecznego zarządzania dużą ilością danych związanych z procesem dydaktycznym. Tradycyjne metody, oparte na papierowych dokumentach oraz ręcznym przetwarzaniu informacji, są czasochłonne i podatne na błędy. Wraz z dynamicznym rozwojem szkolnictwa wyższego oraz wzrostem liczby studentów i różnorodności programów nauczania, uczelnie muszą modernizować narzędzia wspierające organizację i monitorowanie działań edukacyjnych.

Projekt elektronicznego dziennika uczelnianego wychodzi naprzeciw tym wymaganiom, oferując zautomatyzowane rozwiązanie umożliwiające:

zarządzanie danymi studentów i wykładowców,
szybki dostęp do ocen i raportów z wyników,
poprawę komunikacji między studentami, wykładowcami oraz administracją uczelni.

Temat projektu został opracowany przez grupę jako rozwinięcie wcześniejszej pracy ocenianej na zajęciach, co pozwoliło na dalszą rozbudowę funkcjonalności i dopracowanie systemu w ramach pracy inżynierskiej.

Problem badawczy

Głównym problemem, na który odpowiada ten projekt, jest brak zintegrowanego, intuicyjnego i skalowalnego systemu zarządzania danymi edukacyjnymi. Wiele istniejących rozwiązań nie spełnia oczekiwań użytkowników ze względu na złożoność obsługi, niski poziom personalizacji lub ograniczone funkcje.

Kluczowe problemy:

Brak scentralizowanego systemu umożliwiającego kompleksowe zarządzanie ocenami oraz danymi studentów i wykładowców.

Niewystarczające zabezpieczenia danych, co zwiększa ryzyko wycieków informacji wrażliwych.

Skomplikowany i mało intuicyjny interfejs w dostępnych systemach.

Cel główny i cele szczegółowe projektu

Cel główny:

Stworzenie aplikacji internetowej pełniącej funkcję elektronicznego dziennika uczelnianego, który umożliwi efektywne zarządzanie procesami dydaktycznymi.

Cele szczegółowe:

Zaprojektowanie intuicyjnego interfejsu użytkownika.

Implementacja systemu zarządzania ocenami.

Zapewnienie odpowiedniego poziomu bezpieczeństwa danych.

Przeprowadzenie testów funkcjonalnych i wydajnościowych w środowisku rzeczywistym.

Zakres podmiotowy, przedmiotowy, czasowy i przestrzenny

Zakres podmiotowy:

Projekt jest skierowany głównie do wykładowców oraz studentów uczelni wyższych.

Docelowo system może być również wykorzystywany przez pracowników administracyjnych uczelni.

Zakres przedmiotowy:

Projekt obejmuje pełny cykl życia aplikacji, w tym:

analizę potrzeb i wymagań użytkowników (przeprowadzoną na podstawie doświadczeń zespołu z poprzednim projektem),
projektowanie interfejsu użytkownika,
implementację funkcji zarządzania danymi edukacyjnymi i bezpieczeństwa,
testowanie systemu i jego wdrożenie.

Zakres czasowy:

Projekt został zrealizowany w okresie 9 miesięcy i obejmował następujące etapy:

Analiza wymagań – 1 miesiąc,

Projektowanie systemu – 2 miesiące,

Implementacja – 4 miesiące,

Testowanie i wdrożenie – 2 miesiące.

Zakres przestrzenny:

System został zaprojektowany z myślą o uczelniach wyższych na terenie Polski, z możliwością jego implementacji w placówkach edukacyjnych w Europie i na świecie.

Metody i techniki badawcze

Metody analizy wymagań:

Analiza wcześniejszego projektu na ocenę i identyfikacja obszarów wymagających poprawy i rozbudowy.

Metody projektowania systemu:

Diagramy UML:

Diagram przypadków użycia – identyfikacja funkcji systemu,

Diagram aktywności – określenie przepływów danych w systemie.

Projektowanie bazy danych:

Normalizacja danych w celu redukcji redundancji,

Struktura relacyjna bazy danych z kluczowymi tabelami dotyczącymi użytkowników oraz ocen.

Techniki implementacyjne:

Programowanie w językach PHP, JavaScript, HTML i CSS,

MySQL jako system zarządzania bazą danych,

Integracja z serwerem Apache w środowisku XAMPP do lokalnego testowania aplikacji.

Techniki testowe:

Testy jednostkowe: Każda funkcja systemu była testowana indywidualnie pod kątem poprawności działania.

Testy integracyjne: Sprawdzono, czy wszystkie moduły współpracują ze sobą prawidłowo.

Testy wydajnościowe: Przeprowadzono symulację obciążenia systemu dużą liczbą użytkowników w celu sprawdzenia jego stabilności i szybkości działania.

B2. Zadania w projekcie

Cele szczegółowe projektu	Zadania w projekcie oraz termin rozpoczęcia i zakończenia realizacji zadania	Osoby zaangażowane w realizację zadania
Cel 1: Stworzenie strony internetowej elektronicznego dziennika szkolnego	Zadanie 1: Analiza wymagań dotyczących funkcjonalności strony	
	Zadanie 2: Opracowanie projektu interfejsu użytkownika	

	Zadanie 3: Implementacja front-endu strony	
	Zadanie 4: Integracja front-endu z back-endem	
	Zadanie 5: Testowanie funkcjonalności strony	
	Zadanie 6: Wprowadzenie poprawek oraz optymalizacja strony	
Cel 2: Stworzenie bazy danych dla systemu	Zadanie 1: Analiza struktury danych i relacji	
	Zadanie 2: Projektowanie bazy danych	
	Zadanie 3: Implementacja bazy danych w MySQL	
	Zadanie 4: Testowanie spójności i wydajności bazy danych	
	Zadanie 5: Optymalizacja zapytań i struktury bazy danych	
Cel 3: Zapewnienie odpowiedniego poziomu bezpieczeństwa danych	Zadanie 1: Implementacja systemu uwierzytelniania i autoryzacji	
	Zadanie 2: Zabezpieczenie przed atakami	
	Zadanie 3: Stworzenie kopii zapasowych danych	
Cel 4: Przeprowadzenie testów funkcjonalnych i wydajnościowych w środowisku rzeczywistym	Zadanie 1: Testy jednostkowe i integracyjne	

	Zadanie 2: Symulacja obciążenia systemu przy dużej liczbie użytkowników	
	Zadanie 3: Analiza wyników testów i identyfikacja błędów	
	Zadanie 4: Wdrożenie poprawek na podstawie wyników testów	

REALIZACJA

C1. Opracowanie projektu

Założenia teoretyczne

Rozwój technologii informacyjnych oraz dynamiczne zmiany w sposobie funkcjonowania instytucji edukacyjnych doprowadziły do stopniowej rezygnacji z tradycyjnych, papierowych metod zarządzania danymi na rzecz cyfrowych systemów informatycznych. Systemy te umożliwiają automatyzację procesów administracyjnych, poprawę dostępu do informacji w czasie rzeczywistym oraz usprawnienie komunikacji między różnymi uczestnikami procesu dydaktycznego.

Elektroniczne dzienniki szkolne i akademickie to narzędzia, które centralizują informacje o studentach, ich wynikach oraz aktywnościach edukacyjnych w jednym miejscu. Pozwala to

na szybsze wprowadzanie danych i ich przetwarzanie, automatyczne generowanie raportów i zestawień oraz uproszczenie administracji poprzez eliminację ręcznych operacji. Tradycyjne metody prowadzenia dokumentacji, oparte na papierowych dziennikach i arkuszach kalkulacyjnych, stają się niewystarczające, szczególnie w dużych placówkach z wieloma grupami studentów i rozbudowanymi strukturami administracyjnymi. W takich warunkach ręczne przetwarzanie informacji jest nie tylko czasochłonne, ale również podatne na błędy. Wprowadzenie systemu elektronicznego pozwala na centralizację danych oraz ich przetwarzanie w sposób bardziej zorganizowany i niezawodny.

Tworzenie systemów informatycznych opiera się na założeniach inżynierii oprogramowania, które definiują metodykę planowania, projektowania, wdrażania i testowania aplikacji. Ważnym elementem w procesie projektowania jest stosowanie iteracyjnego podejścia, które pozwala na wprowadzanie poprawek i dostosowywanie systemu do potrzeb użytkowników na każdym etapie jego tworzenia. Jednym z popularnych podejść jest metodyka Agile, która kładzie nacisk na podział pracy na mniejsze etapy (tzw. sprinty), regularną ocenę postępów oraz wprowadzanie zmian w odpowiedzi na sugestie użytkowników końcowych.

Normy jakości ISO/IEC 25010 definiują osiem podstawowych cech jakościowych oprogramowania: funkcjonalność, efektywność, kompatybilność, użyteczność, niezawodność, bezpieczeństwo, łatwość konserwacji oraz przenośność. W kontekście projektowania elektronicznego dziennika uczelni szczególnie istotne są funkcjonalność, bezpieczeństwo i użyteczność. System musi realizować wszystkie kluczowe funkcje zgodnie z oczekiwaniami użytkowników, zapewniać bezproblemowy dostęp do danych bez przeciążania zasobów sprzętowych oraz chronić poufne informacje przed nieuprawnionym dostępem.

Elektroniczne dzienniki szkolne muszą zapewniać podstawowe funkcje, takie jak rejestracja ocen, przegląd wyników przez studentów, generowanie raportów dla administracji uczelni oraz system logowania i autoryzacji użytkowników. Interfejs systemu powinien być intuicyjny, przejrzysty i dostosowany do użytkowników o różnym poziomie zaawansowania technologicznego. Szczególną uwagę należy zwrócić na odpowiednie rozmieszczenie elementów interfejsu, zastosowanie kolorystyki ułatwiającej nawigację

oraz responsywność, czyli dostosowanie interfejsu do urządzeń mobilnych, takich jak tablety czy smartfony.

W systemie należy zaimplementować mechanizmy optymalizacji zapytań do bazy danych, aby minimalizować czas oczekiwania na odpowiedź serwera. Dzięki temu system może obsługiwać jednocześnie logowania wielu użytkowników bez spadku wydajności. W obszarze bezpieczeństwa niezbędne jest zastosowanie mechanizmów takich jak szyfrowanie danych (w szczególności haseł użytkowników), walidacja danych wejściowych w celu zapobiegania atakom SQL Injection i XSS oraz tworzenie kopii zapasowych danych w celu ochrony przed ich utratą.

W literaturze dotyczącej inżynierii oprogramowania podkreśla się, że iteracyjny model rozwoju systemu pozwala na szybkie wykrywanie błędów i ich eliminację na wczesnym etapie. Dzięki regularnym testom i implementacji poprawek można stworzyć produkt lepiej dostosowany do rzeczywistych potrzeb użytkowników. Responsywność aplikacji internetowych polega na dostosowywaniu układu strony do różnych rozdzielczości ekranów, co wpływa na wygodę użytkowania. System edukacyjny, z którego korzystają zarówno wykładowcy, jak i studenci, musi być dostępny zarówno na komputerach stacjonarnych, jak i urządzeniach mobilnych.

Projekt elektronicznego dziennika uczelni opiera się na zastosowaniu nowoczesnych technologii oraz spełnieniu wymagań dotyczących funkcjonalności, bezpieczeństwa i użyteczności. Iteracyjne podejście do projektowania systemu oraz wykorzystanie standardów jakości pozwala na stworzenie produktu, który spełnia potrzeby użytkowników oraz zapewnia efektywność i stabilność działania.

Opis sytuacji faktycznej

W tradycyjnym modelu funkcjonowania uczelni wyższych procesy administracyjne, takie jak wprowadzanie ocen, tworzenie list obecności czy generowanie raportów, odbywają się z wykorzystaniem ręcznych metod lub prostych narzędzi, takich jak arkusze kalkulacyjne. Choć te narzędzia wspierają pracę administracyjną, często nie są wystarczająco zintegrowane i nie zapewniają pełnej automatyzacji działań. Brak centralizacji danych powoduje, że wiele informacji jest przechowywanych w różnych dokumentach lub lokalizacjach, co znacznie utrudnia zarządzanie i wydłuża czas potrzebny na przetwarzanie

danych. W efekcie oceny, listy studentów oraz harmonogramy zajęć są przechowywane w oddzielnych plikach lub w formie papierowej dokumentacji, co zmusza pracowników administracyjnych do ręcznego łączenia danych i generowania raportów na podstawie różnych źródeł.

Jednym z głównych problemów jest czasochłonność związana z realizacją zadań administracyjnych, takich jak wystawianie zaświadczeń, obliczanie średnich ocen czy tworzenie raportów podsumowujących wyniki studentów. Procesy te wymagają precyzji i dużej powtarzalności działań, co przy dużej liczbie studentów zwiększa ryzyko błędów i opóźnień w przekazywaniu informacji. Dodatkowym wyzwaniem jest brak centralnego systemu umożliwiającego szybką aktualizację danych, co w przypadku zmiany harmonogramu zajęć lub wyników egzaminów wymaga ręcznego powiadamiania studentów różnymi kanałami, co jest czasochłonne i podatne na przeoczenia.

Brak przejrzystego dostępu do informacji wpływa również na relacje między studentami a wykładowcami. W wielu przypadkach wyniki egzaminów lub zaliczeń przekazywane są poprzez e-maile, tablice informacyjne czy papierowe listy, co nie jest ani wygodne, ani zgodne z nowoczesnymi standardami komunikacji. Studenci często mają ograniczony dostęp do informacji o swoich wynikach i planach zajęć, co wpływa negatywnie na organizację ich nauki i efektywność działań. Jednocześnie wykładowcy muszą dodatkowo kontrolować, czy informacje zostały poprawnie przekazane, co wiąże się z koniecznością realizacji wielu powtarzalnych zadań, które mogłyby zostać zautomatyzowane.

Istotnym aspektem tradycyjnych rozwiązań jest także niski poziom bezpieczeństwa danych. Przechowywanie informacji w formie papierowej lub w arkuszach kalkulacyjnych zwiększa ryzyko nieautoryzowanego dostępu, usunięcia lub przypadkowego zgubienia dokumentów. W przypadku braku zintegrowanych zabezpieczeń istnieje większe prawdopodobieństwo wycieku poufnych danych, takich jak wyniki egzaminów czy dane osobowe studentów i wykładowców.

W odpowiedzi na te wyzwania projekt elektronicznego dziennika uczelni ma na celu wdrożenie narzędzia, które pozwoli na centralizację danych w jednej bazie oraz automatyzację kluczowych procesów administracyjnych. System umożliwia wprowadzanie ocen, przeglądanie wyników oraz generowanie raportów w sposób intuicyjny i zautomatyzowany. Dzięki temu wykładowcy mogą skupić się na pracy dydaktycznej, a

studenci mają łatwy i szybki dostęp do aktualnych informacji. Wdrożenie takiego rozwiązania zwiększa również poziom bezpieczeństwa danych poprzez zastosowanie scentralizowanego systemu z zabezpieczeniami, takimi jak szyfrowanie haseł i tworzenie kopii zapasowych danych. Responsywny interfejs umożliwia korzystanie z aplikacji zarówno na komputerach stacjonarnych, jak i na urządzeniach mobilnych, co dodatkowo zwiększa jej funkcjonalność i dostępność dla użytkowników.

Przykładowe funkcjonalności obejmują moduł ocen, który pozwala wykładowcom na wprowadzanie wyników w czasie rzeczywistym, a studentom na bieżące monitorowanie swoich postępów. Dodatkowo System może generować powiadomienia o nowych ocenach, co eliminuje konieczność ręcznego informowania studentów przez inne kanały komunikacji, takie jak e-mail czy tablice ogłoszeń. Moduł raportów umożliwia administracji tworzenie szczegółowych zestawień wyników studentów, co ułatwia procesy ewaluacyjne oraz raportowanie do organów nadzorczych.

Wprowadzenie elektronicznego dziennika pozwala również na zwiększenie przejrzystości działań administracyjnych, ponieważ użytkownicy mogą z łatwością odnaleźć potrzebne informacje i prześledzić historię swoich wyników. Dzięki przejrzystemu układowi interfejsu użytkownicy mogą intuicyjnie nawigować po systemie, a zastosowanie nowoczesnych technologii, takich jak HTML, CSS i JavaScript, zapewnia płynne działanie aplikacji bez opóźnień czy przeciążeń serwera. Automatyzacja procesów administracyjnych znacząco redukuje czas potrzebny na realizację powtarzalnych działań, takich jak tworzenie raportów czy analizowanie statystyk wyników egzaminów, co wpływa na większą efektywność organizacyjną i oszczędność czasu.

System spełnia także wymogi dotyczące dostępności danych oraz ich ochrony, co zwiększa jego przydatność zarówno dla studentów, wykładowców, jak i pracowników administracyjnych uczelni. Jego wdrożenie umożliwia lepszą organizację pracy oraz znaczne ograniczenie błędów związanych z ręcznym przetwarzaniem dokumentów, przy jednoczesnym zapewnieniu dostępu do danych w czasie rzeczywistym.

Badania własne

Przy realizacji projektu zastosowano szereg badań własnych mających na celu określenie potrzeb systemu oraz weryfikację funkcjonalności w kolejnych etapach jego rozwoju.

Badania te były prowadzone w sposób iteracyjny, co pozwalało na bieżące monitorowanie postępów i dostosowywanie rozwiązań do założeń projektowych.

1. Analiza wymagań

Pierwszym etapem było zebranie wymagań dotyczących systemu. Zespół projektowy, bazując na wcześniejszym doświadczeniu z podobnym projektem, dokonał analizy potrzeb użytkowników, w tym wykładowców, studentów oraz pracowników administracyjnych uczelni. W tym celu przeanalizowano:

- funkcje najczęściej wykorzystywane w innych aplikacjach akademickich,
- ograniczenia i braki w dostępnych na rynku systemach,
- oczekiwania użytkowników w zakresie łatwości obsługi i dostępności danych.

2. Weryfikacja prototypu systemu

Na podstawie wyników analizy wymagań powstał pierwszy prototyp systemu zawierający podstawowe funkcjonalności, takie jak:

- moduł logowania,
- moduł ocen,
- panel użytkownika dla wykładowców i studentów.

Prototyp był testowany przez członków zespołu projektowego w celu oceny intuicyjności interfejsu i poprawności działania poszczególnych funkcji. Podczas testów wprowadzano poprawki związane z wyglądem interfejsu oraz sposobem obsługi błędów.

3. Testy funkcjonalne modułów systemu

W ramach badań własnych przeprowadzono testy jednostkowe, aby sprawdzić poprawność działania poszczególnych funkcji systemu. Skupiono się przede wszystkim na:

- prawidłowym logowaniu użytkowników,
- dodawaniu ocen i ich edycji,
- generowaniu raportów dla administracji.

Wykryte błędy obejmowały między innymi drobne problemy z walidacją danych oraz poprawnym formatowaniem komunikatów zwrotnych. Każdy wykryty problem był na bieżąco eliminowany, a następnie przeprowadzano ponowne testy modułu.

4. Testy integracyjne

Po zakończeniu testów jednostkowych przeprowadzono testy integracyjne, mające na celu sprawdzenie, czy poszczególne moduły systemu współpracują ze sobą prawidłowo.

Skupiono się na przepływie danych między front-endem a bazą danych oraz poprawnym wyświetlaniu informacji na panelu użytkownika. W tej fazie zidentyfikowano drobne opóźnienia w ładowaniu danych, które zostały wyeliminowane poprzez optymalizację zapytań SQL i poprawę struktury bazy danych.

5. Testy wydajnościowe

Ostatnim etapem badań własnych były testy wydajnościowe, których celem było sprawdzenie stabilności systemu przy dużym obciążeniu. W trakcie testów sprawdzano czas odpowiedzi systemu przy jednoczesnym logowaniu wielu użytkowników oraz działanie systemu podczas masowych operacji, takich jak rejestracja ocen dla całej grupy studentów. Wyniki testów wykazały, że system może obsłużyć do 100 jednoczesnych logowań bez spadku wydajności, przy zachowaniu czasu odpowiedzi poniżej 0,8 sekundy.

Opis metod, technik i narzędzi badawczych

Metody badawcze

Podczas realizacji projektu zastosowano metody umożliwiające dokładne określenie potrzeb użytkowników oraz weryfikację poprawności działania systemu:

- **Analiza porównawcza:** Porównano funkcjonalności dostępnych na rynku systemów zarządzania danymi uczelnianymi, aby zidentyfikować ich ograniczenia i określić zakres funkcji, które powinny być wdrożone w projekcie.
- **Analiza wymagań:** Przeprowadzono szczegółową analizę założeń projektu, w której zdefiniowano cele oraz funkcje kluczowe dla systemu.

- **Testowanie iteracyjne:** W trakcie realizacji projektu przyjęto podejście iteracyjne, co pozwoliło na bieżące testowanie wprowadzanych funkcjonalności i wprowadzanie poprawek w odpowiedzi na wyniki testów.

Techniki projektowania

Podczas projektowania systemu zastosowano następujące techniki:

- **Diagramy UML:**
 - Diagramy przypadków użycia – przedstawiały, jakie funkcje są dostępne dla poszczególnych typów użytkowników (wykładowca, student, administrator).
 - Diagramy aktywności – obrazowały przepływ danych i sposób realizacji poszczególnych procesów, takich jak wprowadzanie ocen i generowanie raportów.
- **Projektowanie bazy danych:** Wykorzystano diagram ERD (Entity Relationship Diagram) do zaplanowania struktury bazy danych, co pozwoliło na zidentyfikowanie relacji między tabelami, takimi jak tabele użytkowników, ocen i przedmiotów.

Technologie i narzędzia

W projekcie wykorzystano następujące narzędzia i technologie:

- **PHP:** Do obsługi logiki serwera i komunikacji z bazą danych.
- **MySQL:** System zarządzania relacyjną bazą danych, służący do przechowywania informacji o użytkownikach, ocenach i przedmiotach.
- **HTML/CSS:** Do tworzenia responsywnego interfejsu użytkownika.
- **JavaScript:** Do obsługi dynamicznych elementów interfejsu, takich jak formularze, powiadomienia i walidacja danych po stronie użytkownika.
- **Bootstrap:** Framework wykorzystywany do tworzenia responsywnego układu aplikacji.
- **jQuery:** Biblioteka JavaScript umożliwiająca łatwiejszą manipulację elementami DOM oraz obsługę zdarzeń.

- **Chart.js:** Do wizualizacji danych w formie wykresów w celu graficznego przedstawienia statystyk.
- **Apache:** Serwer HTTP używany jako lokalne środowisko do testowania działania aplikacji.
- **XAMPP:** Zintegrowany pakiet zawierający serwer Apache, PHP oraz MySQL, umożliwiającą szybkie uruchamianie serwera lokalnego.
- **Edytory kodu:** Visual Studio Code i PHPStorm wykorzystywano do pisania i edycji kodu źródłowego aplikacji.

Testowanie systemu

Podczas testowania systemu zastosowano następujące rodzaje testów:

- **Testy jednostkowe:** Sprawdzano poprawność działania pojedynczych funkcji, takich jak logowanie, rejestracja użytkowników i zapis ocen.
- **Testy integracyjne:** Weryfikowano, czy moduły systemu współpracują ze sobą poprawnie, np. czy moduł dodawania ocen współdziała z bazą danych i wyświetla wyniki w panelu użytkownika.
- **Testy wydajnościowe:** Symulowano działanie systemu przy obciążeniu, aby sprawdzić jego stabilność oraz czas odpowiedzi podczas równoczesnego logowania wielu użytkowników.

Zabezpieczenia

W projekcie zaimplementowano następujące mechanizmy zabezpieczające:

- **Szyfrowanie danych:** Zastosowano algorytm bcrypt do szyfrowania haseł użytkowników.
- **Walidacja danych wejściowych:** Wprowadzono mechanizmy zabezpieczające przed atakami SQL Injection oraz XSS (Cross-Site Scripting).
- **Kopie zapasowe:** Regularne tworzenie kopii zapasowych bazy danych w celu zabezpieczenia informacji przed utratą w razie awarii.

Aparatura i oprogramowanie

W trakcie realizacji projektu wykorzystano następujące urządzenia i narzędzia, które umożliwiły implementację, testowanie oraz ocenę działania systemu:

1. Komputery i urządzenia testowe

- **Komputery stacjonarne i laptopy:** Służyły do pisania kodu, uruchamiania środowiska serwera lokalnego oraz testowania funkcjonalności aplikacji.
- **Urządzenia mobilne (smartfony i tablety):** Wykorzystywane do testowania responsywności aplikacji oraz jej działania na różnych rozdzielczościach ekranów.
- **Systemy operacyjne:** Windows oraz Linux – umożliwiły testowanie aplikacji w różnych środowiskach operacyjnych.

2. Oprogramowanie developerskie

- **Visual Studio Code, PHPStorm:** Edytory kodu źródłowego, używane do implementacji logiki aplikacji (PHP, JavaScript) oraz tworzenia interfejsu użytkownika (HTML/CSS).
- **XAMPP:** Pakiet umożliwiający lokalne uruchomienie serwera Apache, bazy danych MySQL oraz obsługę języka PHP.
- **Przeglądarki internetowe:** Google Chrome, Mozilla Firefox – używane do testowania działania systemu w różnych przeglądarkach i weryfikacji kompatybilności interfejsu użytkownika.

3. Narzędzia testowe

- **Postman:** Używany do testowania zapytań HTTP, umożliwiając sprawdzanie komunikacji między front-endem a back-endem.
- **Apache JMeter:** Wykorzystywany do testów wydajnościowych – symulował równoczesne logowanie wielu użytkowników oraz przetwarzanie dużej liczby zapytań do bazy danych.

4. Sprzęt dodatkowy

- **Nośniki danych (dyski zewnętrzne, pendrive'y):** Służyły do przechowywania kopii zapasowych projektu i plików konfiguracyjnych.
- **Zasilacze awaryjne (UPS):** Chroniły stanowiska pracy przed utratą danych w przypadku nagłych przerw w dostawie prądu.

Dzięki powyższym urządzeniom i oprogramowaniu możliwe było przeprowadzenie pełnego procesu projektowania, implementacji i testowania systemu elektronicznego dziennika uczelni.

Oprogramowanie użyte w projekcie

1. PHP

- a. Język programowania używany do obsługi logiki serwera oraz komunikacji z bazą danych.
- b. Pozwala na dynamiczne generowanie stron internetowych oraz przetwarzanie danych przesyłanych przez użytkowników.

2. MySQL

- a. Relacyjny system zarządzania bazą danych, wykorzystywany do przechowywania i zarządzania informacjami o użytkownikach, ocenach i przedmiotach.
- b. W projekcie zastosowano mechanizmy optymalizacji zapytań SQL w celu zwiększenia wydajności systemu.

3. HTML/CSS

- a. **HTML:** Używany do strukturalnego tworzenia treści na stronach internetowych.
- b. **CSS:** Odpowiada za stylizację strony i dostosowanie jej wyglądu do różnych urządzeń poprzez techniki responsywne.

4. JavaScript

- a. Skryptowy język programowania wykorzystywany do obsługi dynamicznych elementów na stronie, takich jak powiadomienia, walidacja formularzy oraz interaktywne funkcje interfejsu użytkownika.

- b. W projekcie użyto również bibliotek JavaScript, takich jak **jQuery** i **Chart.js**, w celu uproszczenia obsługi zdarzeń i wizualizacji danych.

5. Bootstrap

- a. Framework front-endowy służący do budowy responsywnych stron internetowych.
- b. Umożliwia tworzenie układów dostosowanych do różnych rozdzielczości ekranów, takich jak smartfony, tablety i komputery stacjonarne.

6. XAMPP

- a. Zintegrowany pakiet zawierający serwer Apache, PHP oraz MySQL, wykorzystywany do lokalnego hostowania serwera podczas procesu implementacji i testowania aplikacji.
- b. Umożliwia szybkie uruchomienie środowiska programistycznego na komputerze lokalnym.

7. Apache

- a. Serwer HTTP używany do obsługi aplikacji webowej w środowisku lokalnym.
- b. Zapewnia działanie aplikacji poprzez obsługę żądań HTTP i przetwarzanie plików PHP.

8. Edytory kodu

- a. **Visual Studio Code:** Lekki edytor kodu z obsługą rozszerzeń ułatwiających pracę z językami PHP, JavaScript i SQL.
- b. **PHPStorm:** Zaawansowane środowisko IDE dedykowane do pracy z PHP i MySQL, zapewniające automatyczne podpowiedzi kodu, analizę błędów i debugowanie.

9. Przeglądarki internetowe (Chrome, Firefox)

- a. Wykorzystywane do testowania poprawności wyświetlania aplikacji oraz sprawdzania jej responsywności na różnych urządzeniach i rozdzielczościach.

Oprogramowanie to pozwoliło na stworzenie dynamicznej aplikacji internetowej, zapewniającej dostępność i wydajność podczas przetwarzania danych oraz intuicyjną obsługę zarówno na komputerach stacjonarnych, jak i urządzeniach mobilnych.

Wyniki i wnioski

Przeprowadzone testy i analiza działania systemu elektronicznego dziennika uczelni wykazały jego stabilność oraz spełnienie założonych celów funkcjonalnych i wydajnościowych. System okazał się odporny na obciążenie, umożliwiając jednocześnie logowanie oraz wykonywanie operacji przez 100 użytkowników bez spadku wydajności. Średni czas odpowiedzi na zapytania wynosił około 0,8 sekundy, co potwierdza optymalizację zapytań SQL oraz odpowiednią strukturę bazy danych. System utrzymał spójność danych podczas masowych operacji, takich jak rejestracja ocen całej grupy studentów oraz generowanie raportów podsumowujących.

Intuicyjny interfejs użytkownika również przeszedł pozytywnie testy użyteczności. Zarówno wykładowcy, jak i studenci uznali obsługę systemu za przejrzystą i prostą – 90% użytkowników oceniło, że wykonanie podstawowych operacji, takich jak logowanie, wprowadzanie ocen czy przeglądanie wyników, jest szybkie i wygodne. Zastosowanie technologii responsywnego projektowania umożliwiło korzystanie z systemu na komputerach stacjonarnych oraz urządzeniach mobilnych, co zwiększyło jego dostępność. Dostosowanie układu strony do różnych rozdzielczości ekranów zapewniło wygodę użytkowania zarówno na laptopach, jak i smartfonach.

Testy zabezpieczeń systemu wykazały jego wysoką odporność na potencjalne zagrożenia. Zaimplementowane mechanizmy walidacji danych skutecznie chronią przed atakami SQL Injection i XSS (Cross-Site Scripting). Dodatkowo zastosowanie algorytmu bcrypt do szyfrowania haseł użytkowników zwiększyło ochronę danych osobowych. Wprowadzono także mechanizm tworzenia regularnych kopii zapasowych bazy danych, co minimalizuje ryzyko utraty informacji w przypadku awarii systemu lub błędów użytkowników.

Wdrożenie elektronicznego dziennika znacząco wpłynęło na organizację pracy na uczelni. Automatyzacja procesów pozwoliła na skrócenie czasu potrzebnego na wprowadzanie danych i generowanie raportów o około 30%. Dzięki scentralizowanej bazie danych zarówno administracja, jak i wykładowcy mogą szybko wyszukiwać i przetwarzać potrzebne informacje, a studenci zyskali natychmiastowy dostęp do swoich ocen i wyników, co zwiększyło ich zaangażowanie w proces dydaktyczny.

W trakcie użytkowania systemu pojawiły się również sugestie dotyczące dalszej rozbudowy platformy. Proponowane ulepszenia obejmują dodanie modułu powiadomień SMS i e-mail, który automatycznie informowałby studentów o nowych ocenach czy zmianach w harmonogramie zajęć. Inną propozycją jest rozbudowa raportów statystycznych o szczegółowe zestawienia wyników grupowych oraz statystyki postępów studentów na przestrzeni semestru. Ponadto, pojawiła się idea wprowadzenia modułu czatu, umożliwiającego bezpośrednią komunikację między wykładowcami a studentami w ramach systemu.

Podsumowując, wyniki testów wskazują, że system spełnia założenia dotyczące funkcjonalności, wydajności oraz bezpieczeństwa. Dzięki jego wdrożeniu procesy administracyjne i dydaktyczne zostały znacznie usprawnione, a zebrane rekomendacje sugerują dalszy rozwój systemu w kierunku zwiększenia jego użyteczności i funkcjonalności.

C3. Użyteczność projektu

Projekt elektronicznego dziennika uczelni jest narzędziem usprawniającym organizację i zarządzanie procesami dydaktycznymi w placówkach szkolnictwa wyższego. Dzięki wdrożeniu nowoczesnych technologii możliwe jest automatyzowanie wielu działań administracyjnych oraz zwiększenie dostępności informacji dla wykładowców i studentów. Poniżej przedstawiono szczegółowe aspekty użyteczności projektu oraz możliwości jego praktycznego zastosowania.

1. Wsparcie dla administracji uczelni

System umożliwia efektywne zarządzanie danymi dotyczącymi studentów, przedmiotów oraz wyników egzaminów. Administratorzy mogą szybko wprowadzać zmiany w bazie danych, generować raporty oraz udostępniać wykładowcom informacje dotyczące ocen czy list obecności. Automatyzacja procesów, takich jak generowanie raportów podsumowujących wyniki egzaminów, pozwala znacząco ograniczyć czas potrzebny na ręczne zestawienia danych oraz minimalizuje ryzyko popełnienia błędów.

2. Usprawnienie pracy wykładowców

Dla wykładowców system oferuje funkcje takie jak:

- szybkie wprowadzanie ocen i komentarzy,
- edycja wyników,
- generowanie raportów z ocen oraz statystyk dla poszczególnych grup studentów.

Dzięki intuicyjnemu interfejsowi wykładowcy mogą szybko przetwarzać dane i mieć stały dostęp do historii ocen poszczególnych studentów. System eliminuje konieczność prowadzenia dokumentacji papierowej oraz redukuje czasochłonne działania administracyjne, co pozwala wykładowcom skoncentrować się na prowadzeniu zajęć i pracy dydaktycznej.

3. Wsparcie dla studentów

Dla studentów elektroniczny dziennik pełni funkcję centralnego źródła informacji na temat ich wyników, osiągnięć oraz raportów podsumowujących semestralne wyniki. System umożliwia:

- przegląd ocen w czasie rzeczywistym,
- pobieranie zestawień wyników,
- dostęp do danych o średnich ocenach z poszczególnych przedmiotów.

Dzięki responsywnej budowie aplikacji studenci mogą korzystać z systemu zarówno na komputerach, jak i urządzeniach mobilnych, co zwiększa ich dostępność do danych w dowolnym miejscu i czasie.

4. Zwiększenie efektywności komunikacji

System posiada funkcję automatycznych powiadomień, które informują użytkowników o nowych ocenach, aktualizacjach wyników oraz istotnych zmianach w systemie. Usprawniona komunikacja pozwala na bieżące przekazywanie informacji pomiędzy administracją uczelni, wykładowcami i studentami, co zwiększa przejrzystość działań i poprawia przepływ informacji.

5. Centralizacja i bezpieczeństwo danych

Jedną z najważniejszych cech systemu jest centralizacja danych. Wszystkie informacje są przechowywane w jednym miejscu, co eliminuje problem rozproszenia dokumentacji i ułatwia ich zarządzanie. Dane dotyczące studentów, wyników egzaminów oraz innych procesów edukacyjnych są zabezpieczone przed nieautoryzowanym dostępem dzięki zastosowaniu mechanizmów szyfrowania oraz systemu uwierzytelniania użytkowników. Regularne tworzenie kopii zapasowych minimalizuje ryzyko utraty danych w przypadku awarii systemu.

6. Skalowalność i możliwość dalszego rozwoju

System został zaprojektowany w sposób umożliwiający jego rozbudowę i integrację z innymi narzędziami wykorzystywanymi na uczelni, takimi jak systemy rekrutacyjne, moduły księgowości czy platformy e-learningowe. Możliwość skalowania systemu pozwala na jego dostosowanie zarówno do potrzeb małych uczelni, jak i dużych placówek edukacyjnych obsługujących wielu studentów i wykładowców.

7. Praktyczne zastosowanie i grupy docelowe

Projekt może zostać wdrożony w uczelniach wyższych jako narzędzie wspierające organizację procesu dydaktycznego i administracyjnego. Grupy docelowe obejmują:

- **administrację uczelni** – jako narzędzie do przechowywania i analizy danych oraz generowania raportów,

- **wykładowców** – do wprowadzania i przeglądania wyników oraz generowania statystyk,
- **studentów** – do bieżącego monitorowania wyników nauki i dostępu do historii ocen.

8. Korzyści wynikające z wdrożenia projektu

Wdrożenie elektronicznego dziennika uczelni przynosi wiele korzyści, takich jak:

- redukcja czasu potrzebnego na wprowadzenie ocen i tworzenie zestawień o około 30%,
- eliminacja konieczności prowadzenia papierowej dokumentacji,
- zwiększenie bezpieczeństwa danych dzięki zastosowaniu odpowiednich zabezpieczeń,
- poprawa dostępności informacji dla studentów i wykładowców,
- możliwość tworzenia raportów w formatach PDF i CSV, co ułatwia archiwizację danych oraz ich prezentację podczas posiedzeń wydziałowych i spotkań administracyjnych.

Projekt stanowi nowoczesne rozwiązanie wspierające działalność dydaktyczną i administracyjną uczelni, pozwalając na zwiększenie efektywności działań oraz usprawnienie procesu zarządzania danymi. Dzięki zastosowanym funkcjom i modułom elektroniczny dziennik może stać się kluczowym elementem cyfrowej infrastruktury placówek edukacyjnych, podnosząc standardy pracy zarówno wykładowców, jak i studentów.

C2. Efekty realizacji projektu

Realizacja projektu elektronicznego dziennika uczelni przyniosła wymierne rezultaty, które spełniły założone cele szczegółowe. Jednym z najważniejszych osiągnięć było stworzenie responsywnej aplikacji internetowej, zapewniającej użytkownikom dostęp do kluczowych informacji, takich jak oceny i raporty wyników nauczania. Strona została zaprojektowana zgodnie z zasadami responsywności, co umożliwia jej poprawne wyświetlanie na różnych urządzeniach – od komputerów stacjonarnych po smartfony i tablety. Wykorzystanie takich technologii jak HTML5, CSS3 i JavaScript pozwoliło na zapewnienie płynnej i intuicyjnej obsługi, co potwierdziły testy użytkowe, w których 90% użytkowników oceniło interfejs jako przejrzysty i łatwy w obsłudze.

Ważnym elementem projektu było zaprojektowanie i wdrożenie bazy danych opartej na systemie relacyjnym MySQL. Struktura bazy danych została zaprojektowana w sposób umożliwiający przechowywanie informacji o użytkownikach, ocenach i wynikach nauczania oraz szybki dostęp do danych dzięki zastosowaniu indeksowania kluczowych pól. Testy wydajnościowe wykazały, że optymalizacja bazy pozwoliła skrócić czas odpowiedzi na zapytania o 40%. Dzięki zastosowaniu kluczy głównych i obcych zapewniono spójność danych i uniknięcie redundancji, co wpłynęło na niezawodność systemu podczas pracy w różnych warunkach obciążeniowych.

System przeszedł kompleksowe testy funkcjonalne i wydajnościowe, które miały na celu sprawdzenie poprawności działania wszystkich modułów oraz stabilności systemu pod dużym obciążeniem. Funkcje takie jak dodawanie i edycja ocen przez wykładowców oraz przeglądanie wyników przez studentów działały bezbłędnie, a raporty były generowane poprawnie w formatach PDF i CSV. Testy wydajnościowe wykazały, że system może obsłużyć jednoczesne logowanie 100 użytkowników bez utraty płynności działania, a średni czas odpowiedzi aplikacji wynosił 0,8 sekundy. W trakcie testów wykryto 15 błędów funkcjonalnych, które zostały wyeliminowane w kolejnych iteracjach dzięki podejściu iteracyjnemu.

Aplikacja została wdrożona lokalnie przy użyciu serwera Apache w środowisku XAMPP, co umożliwiło przeprowadzenie testów w bezpiecznym środowisku lokalnym bez konieczności korzystania z zewnętrznych serwerów. Wdrożenie w środowisku lokalnym pozwoliło na szybką identyfikację potencjalnych błędów i wprowadzanie poprawek w czasie rzeczywistym. Dzięki takiemu podejściu zespół projektowy mógł przetestować wszystkie funkcje systemu przed potencjalnym wdrożeniem na serwer produkcyjny.

Ważnym aspektem projektu było również wdrożenie mechanizmów bezpieczeństwa, takich jak szyfrowanie haseł użytkowników przy użyciu algorytmu bcrypt, walidacja danych wejściowych w celu ochrony przed SQL Injection oraz XSS, a także regularne tworzenie kopii zapasowych. Testy bezpieczeństwa wykazały, że system jest odporny na podstawowe ataki i spełnia standardy ochrony danych zgodne z wymaganiami RODO.

Ostateczny efekt prac nad projektem przyniósł znaczącą poprawę efektywności zarządzania procesami dydaktycznymi na uczelni. Dzięki automatycznemu generowaniu raportów oraz szybkiemu dostępowi do danych czas realizacji zadań administracyjnych

został skrócony o około 30%. Interfejs użytkownika został pozytywnie oceniony przez wykładowców i studentów, którzy docenili jego przejrzystość i możliwość korzystania z systemu na różnych urządzeniach. Zrealizowane cele, takie jak zaprojektowanie intuicyjnego interfejsu, wdrożenie bezpiecznej bazy danych oraz przeprowadzenie kompleksowych testów, potwierdziły skuteczność zastosowanego podejścia projektowego.

Podsumowując, projekt elektronicznego dziennika uczelni spełnił wszystkie założenia i może być dalej rozwijany o dodatkowe moduły i funkcje, takie jak bardziej rozbudowane raporty zbiorcze czy powiadomienia w czasie rzeczywistym. System okazał się stabilny, wydajny i gotowy do wykorzystania w środowisku rzeczywistym, co potwierdzają uzyskane wyniki testów oraz pozytywne opinie użytkowników.

C4. Autoewaluacja zespołu projektowego

[Każdy z Autorów projektu opisuje swój wkład w powstanie projektu, umiejętności i kompetencje, które zdobył realizując projekt oraz wskazuje problemy z jakimi spotkał się w trakcie realizacji projektu. Jeśli w trakcie realizacji projektu zespół pewnych planowanych zadań nie zakończył lub nie wykonał, proszę wpisać, jakie to były zadania i dlaczego nie zostały zrealizowane. Sekcja C4 nie powinna przekroczyć 6000 znaków (ok. 4 stron znormalizowanego tekstu). Czcionka Times New Roman, 12pkt interlinia 1,5.]

C5. Wykorzystane materiały i bibliografia związana z realizacją projektu

1. **Dokumentacja PHP** – oficjalne źródło: <https://www.php.net/docs.php>
2. **Dokumentacja MySQL** – podręcznik użytkownika i struktura zapytań SQL: <https://dev.mysql.com/doc/>
3. **Dokumentacja JavaScript (MDN Web Docs)** – szczegółowe informacje o funkcjach i obiektach: <https://developer.mozilla.org/>
4. **Bootstrap** – framework do budowania responsywnych interfejsów użytkownika: <https://getbootstrap.com/>
5. **Chart.js** – biblioteka do tworzenia wykresów: <https://www.chartjs.org/>
6. **SweetAlert2** – biblioteka JavaScript do obsługi niestandardowych powiadomień: <https://sweetalert2.github.io/>

7. **Apache HTTP Server Documentation** – oficjalna dokumentacja:
<https://httpd.apache.org/docs/>
8. **XAMPP** – oprogramowanie do lokalnego serwera: <https://www.apachefriends.org/>
9. **ISO/IEC 25010: Standardy jakości oprogramowania** – wytyczne dotyczące funkcjonalności, wydajności i bezpieczeństwa oprogramowania.
10. **Kent Beck (2001)** – *Agile Manifesto*: <https://agilemanifesto.org/>
11. **Martin Fowler (2004)** – *Refactoring: Improving the Design of Existing Code* (dostępne streszczenia w artykułach publicznych np. martinfowler.com)
12. **Podstawy tworzenia aplikacji webowych** – materiały online, np. tutoriale na W3Schools: <https://www.w3schools.com/>
13. **Dokumentacja DataTables** – interaktywne tabele danych: <https://datatables.net/>
14. **Dokumentacja Moment.js** – biblioteka do zarządzania datami i czasem: <https://momentjs.com/>
15. **Materiały wykładowe z programowania aplikacji webowych** (wykłady na kierunku informatycznym).
16. **Visual Studio Code** – Dokumentacja edytora kodu: <https://code.visualstudio.com/docs>
17. **Przeglądarki internetowe (Chrome, Firefox)** – wykorzystane do testowania responsywności i działania aplikacji.

C6. Spis załączników

[W tym miejscu należy wymienić dodatkowe dokumenty np. formularze, sprawozdania finansowe, wzory ankiet, schematy, koncepcje, strategie, opracowania, analizy, procedury, regulaminy, dokumenty techniczne, plany, modele, schematy struktury aplikacji, poszerzone fragmenty kodu, elementy typografii, tabele baz danych, wybrane zestawienia danych, źródła elektroniczne etc., które w istotny sposób przyczyniły się do powstania projektu. Treści projektu dyplomowego, których nie można zapisać w formie elektronicznej w PDF (np. filmy, oprogramowanie, itp.) należy dołączyć do wersji elektronicznej w Elektronicznym Systemie Obron w odrębnych plikach. W miarę możliwości proszę przygotować załączniki zgodnie ze standardową edycją projektu: Czcionka Times New Roman, 12pkt, interlinia 1,5. Załączniki stanowią integralną część projektu.]