

Collegium Witelona Uczelnia Państwowa w Legnicy

Sprawozdanie z PPSI 1



To Do App

Mikołaj Wołoszyn, nr indeksu 43901

Roksana Żyłka, nr indeksu 43908

Kamil Kubrak, nr indeksu 43855

Rok studiów: II

Rok akademicki 2024/2025

Prowadzący przedmiot: mgr inż. Krzysztof Rewak

Legnica 2025

Streszczenie

Celem tego sprawozdania jest przedstawienie pracy nad zespołowym projektem - aplikacją webową, która pomaga użytkownikom zarządzać zadaniami. Opisujemy tu, co dokładnie system potrafi, jakich technologii użyliśmy i jak wyglądała jego implementacja. Na końcu dzielimy się wnioskami z pracy oraz pomysłami, jak można ten projekt dalej rozwijać.

Spis treści

1	Opis funkcjonalny systemu	2
2	Opis technologiczny	2
3	Wdrożone zagadnienia kwalifikacyjne	3
4	Instrukcja uruchomienia systemu	4
4.1	Lokalne uruchomienie	4
5	Wnioski projektowe	4

1 Opis funkcjonalny systemu

System został zaprojektowany w celu zarządzania zadaniami użytkownika w formie aplikacji webowej. Pozwala na rejestrację, logowanie, dodawanie zadań, zarządzanie profilem. Główne funkcjonalności systemu to:

- Rejestracja i logowanie użytkownika – tworzenie konta i autoryzacja,
- Tworzenie i usuwanie tasków – użytkownicy mogą zarządzać swoją listą zadań,
- Zarządzanie profilem użytkownika - edycja danych i zdjęcia profilowego,
- Zachowanie sesji użytkownika - dostęp chroniony po zalogowaniu.

Interakcja użytkownika z systemem odbywa się poprzez przeglądarkę internetową, w ramach dynamicznych widoków generowanych przez silnik szablonów Thymeleaf.

2 Opis technologiczny

Projekt został zaimplementowany z wykorzystaniem następujących technologii:

- Języki programowania:
 - Java,
 - JavaScript,
 - HTML/CSS.
- Frameworki i biblioteki:
 - Spring Boot - backend i REST API,
 - Thymeleaf - silnik szablonów HTML,
 - Bootstrap - stylowanie interjesu,
 - Vue.js - biblioteka js.
- System zarządzania bazą danych:
 - MySQL;
- Narzędzia developerskie:
 - IntelliJ IDEA - IDE,
 - Git + GitHub - wersjonowanie kodu,
 - Gradle - budowanie projektu.

Struktura systemu składa się z modułów: frontend (szablony HTML + JS), backend (kontrolery, modele, serwisy, repozytoria), konfiguracja bezpieczeństwa oraz zarządzanie sesją/logowaniem.

3 Wdrożone zagadnienia kwalifikacyjne

W ramach projektu zaimplementowano następujące zagadnienia kwalifikacyjne:

1. Framework MVC - zastosowano wzorec architektoniczny Model-View-Controller przy użyciu frameworka Spring Boot, który zarządza przepływem danych i logiką aplikacji,
2. Framework CSS - do stylizacji interfejsu użytkownika wykorzystano Bootstrap, zapewniający spójny i responsywny wygląd,
3. Baza danych - projekt został zintegrowany z relacyjną bazą danych MySQL, w której przechowywane są dane użytkowników i tasków,
4. Cache - zaimplementowano prosty mechanizm buforowania danych w celu przyspieszenia dostępu,
5. Dependency manager - wykorzystano Gradle do zarządzania zależnościami projektu,
6. HTML - struktura aplikacji webowej,
7. CSS - interfejs został ostyleowany przy jego użyciu,
8. JavaScript - dodano elementy interaktywne na stronie, takie jak formularze i dynamiczne komunikaty,
9. Routing - zaimplementowano system routingu w Spring Boot(np. /login, /tasks, /admin),
10. ORM - zastosowano Spring Data JPA do mapowania obiektowo-relacyjnego pomiędzy klasami Javy a tabelami w bazie,
11. Uwierzytelnianie - stworzono system logowania i autoryzacji użytkowników,
12. Lokalizacja - możliwość przełączania z języka polskiego na angielski,
13. Mailing - wysyłanie resetowanego hasła na maila,
14. Formularze - dane przesyłane są do systemu za pomocą formularzy HTML, przetwarzanych przez kontrolery Spring,
15. Asynchroniczne interakcje - część akcji wykonywana jest bez przeładowywania strony,
16. konsumpcja API - użycie SendGrid do resetowania hasła,
17. publikacja API - udostępniono własne REST API umożliwiające integrację z systemem,
18. RWD - responsywny frontend - interfejs przystosowany do działania na różnych rozdzielczościach, dzięki Bootstrapowi,
19. Logger - w kodzie znajdują się elementy logowania akcji, przydatne do monitorowania działania systemu,

4 Instrukcja uruchomienia systemu

4.1 Lokalne uruchomienie

Aby uruchomić system lokalnie:

1. Zainstaluj Javę (JDK 17+), IntelliJ IDEAD (lub inne IDE);
2. Sklonuj repozytorium: https://github.com/xNelion21/projekt_ppsi.git;
3. W IntelliJ IDEAD: otwórz projekt jako "Spring Boot project";
4. Skonfiguruj połączenie z bazą danych;
5. Uruchom projekt (TaskManagerApplication.java);
6. Otwórz przeglądarkę i przejdź do <http://localhost:8080>.

5 Wnioski projektowe

Realizacja projektu pozwoliła na:

- Pogłębienie wiedzy z zakresu frameworka Spring Boot i architektury MVC;
- Rozwój umiejętności praktycznych w zakresie integracji frontend-backend;
- Pracę z relacyjną bazą danych i szablonami Thymeleaf;
- Praktyczne poznanie koonfiguracji systemu logowania.

Potencjalne kierunki rozwoju systemu to:

- Wdrożenie powiadomień,
- Panel statystyk zadań i czasu wykonania,
- Stworzenie grup dla użytkowników, w której będą mogli razem planować,
- Dalsze doskonalenie aplikacji.