

| | |
|---------------------------|---|
| Rok akademicki | 2020/2021 |
| Przedmiot | Projekt zespołowy systemu informatycznego |
| Forma zajęć | Laboratorium |
| Prowadzący zajęcia | Mgr inż. Artur Pala |
| Nr grupy | 6 |
| Sygnatura grupy | P6D3 |
| Kierunek studiów | Informatyka |

Projekt zespołowy systemu informatycznego

„Anielska kuchnia”

Serwis kulinarny

| Nazwisko i imię | Nr indeksu |
|------------------------------------|-------------------|
| Uryga Tomasz – Scrum Master | 97795 |
| Wołożonek Patryk | 98142 |
| Wasiak Damian | 97799 |

| Uwagi |
|-------|
| |

1.Opis projektu

Aplikacja zawierać będzie zbiór przepisów dodawanych przez aktywnych użytkowników. Zostanie ona odpowiednio stylizowana ,dodamy między innymi system dodawania komentarzy, oceny przepisów, umożliwimy logowanie do serwisu za pomocą np. Facebooka . Dodatkowe funkcjonalności to np. wyszukiwanie przepisu, filtrowanie wyników wyszukiwania, modyfikacja własnego profilu, możliwość wyświetlania ulubionych przepisów, obserwacja użytkowników i powiadomienia o ich nowych przepisach.

Jako nasze autorskie pomysły:

- dodamy możliwość wyszukiwania przepisu , za określoną kwotę.
- możliwość określenia kaloryczności posiłku
- system, który generował by jadłospis na dzień w oparciu o preferencje użytkownika

2.Wymagania funkcjonalne

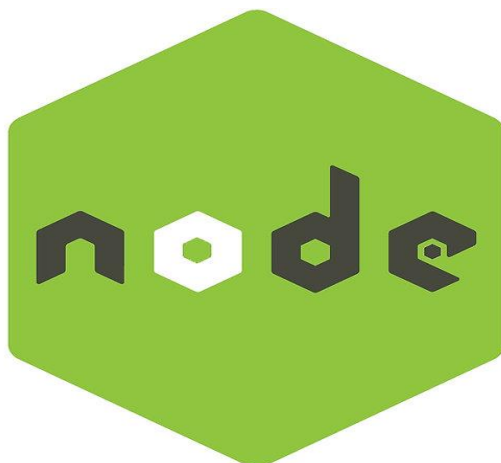
| LP. | Funkcja |
|-----|--|
| 1. | Wyświetlanie przepisów użytkowników |
| 2. | Przypomnienie o nowych przepisach |
| 3. | Dodawanie komentarzy |
| 4. | Wyświetlanie komentarzy |
| 5. | Możliwość dodawania ocen przepisów |
| 6. | Możliwość wyświetlenia przepisu |
| 7. | Integracja z kalendarzem google |
| 8. | Możliwość określenia kaloryczności posiłku |
| 9. | Generowanie jadłospisów w oparciu o preferencje użytkownika |
| 10. | Wyświetlenie rekomendowanych przepisów(top tygodnia) |
| 11. | Możliwość zarejestrowania użytkownika |
| 12. | Możliwość logowania przez zewnętrzne serwisy(Facebook, Google) |
| 13. | Możliwość filtrowania wyników wyszukiwania |
| 14. | Możliwość modyfikacji własnego profilu |
| 15. | Wyświetlanie ulubionych przepisów |

3.Wymagania нефunkcjonalne

- Aplikacja webowa będzie szybka i prosta w użyciu
- Zapewniona pełna responsywność co umożliwi korzystanie z urządzeń mobilnych
- Prosta w utrzymaniu i rozszerzalna(łatwo podlega rozbudowie)
- Zbudowana według architektury MVC

4.Opis środowiska programistycznego

Aplikacja zostanie w bardzo popularnych na rynku technologiach , dotyczących tzw. MERN stacku (MongoDB, Express.js, React.js, Node.js)



Rys.1 Logo technologii Node.js



Rys.2 Logo technologii React.js

React to biblioteka JavaScript stworzona do tworzenia szybkich i interaktywnych interfejsów użytkownika dla aplikacji internetowych i mobilnych. Jest to biblioteka typu Open-source, oparta na komponentach, dla front-endu odpowiedzialna tylko za warstwę widoku aplikacji.

Node.js to wieloplatformowe środowisko uruchomieniowe Open-source do tworzenia aplikacji typu server-side napisanych w języku JavaScript. Natomiast aby zaprogramować aplikację webową musimy dołożyć do tego standardowy framework jakim jest **Express.js**. Jest to struktura serwera aplikacji internetowych opartych na Node, która została specjalnie zaprojektowana do tworzenia jednostronicowych, wielostronicowych i hybrydowych aplikacji internetowych. Stała się ona standardową platformą serwerową szeroko używaną w różnych projektach biznesowych.

MongoDB - otwarty, nierelacyjny system zarządzania bazą danych napisany w języku C++. Charakteryzuje się brakiem ściśle zdefiniowanej struktury

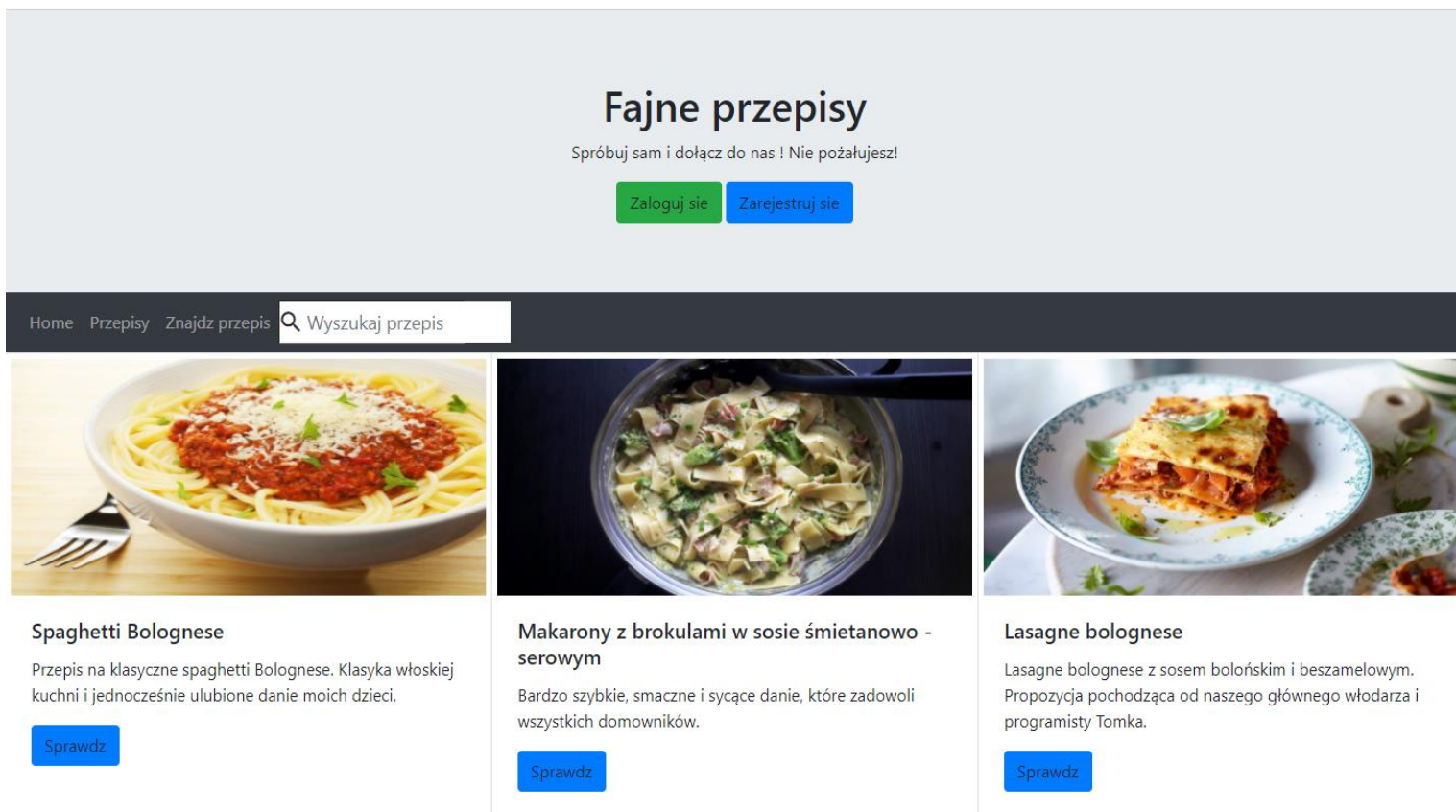
obsługiwanych baz danych. Zamiast tego dane składowane są jako dokumenty w stylu BSON.

Wszystkie powyższe technologie składają się na **MERN Stack**(stos MERN). Jego nazwa pochodzi od pierwszych liter technologii , które są w nim zawarte.

Zalety MERN Stack:

- tworzenie aplikacji webowych używając jednego języka programowania dla zarówno frontendu(React.js) jak i backendu(Express.js)
- ma bardzo duże wsparcie społeczności, zarówno w kwestii rozwiązywania problemów programistycznych (wysoko w rankingach stackoverflow) , jak również zgłaszania błędów (wszystkie komponenty składające się na MERN Stack są open-source)
- wspiera architekturę MVC, którą chcemy wykorzystać
- jest łatwy do implementacji i nauki

5.Grafika koncepcyjna



6.User-stories

Jako użytkownik chce móc przeglądać przepisy, żeby uzyskać informacje na temat interesującego przepisu

Warunki satysfakcji :

- 1.Przepisy wyświetlają się prawidłowo

Jako użytkownik chce móc dodawać komentarze, żeby wyrazić swoją opinie o przepisie

Warunki satysfakcji :

- 1.Komentarz został dodany bezbłędnie.
- 2.Komentarz zostaje prawidłowo wyświetlony

Jako użytkownik chce móc usuwać komentarze, żeby usunąć swoją opinie o przepisie

Warunki satysfakcji :

- 1.Komentarz został usunięty bezbłędnie.
- 2.Komentarz nie wyświetla się .

Jako użytkownik chce móc dodawać opis przepisów, żeby dostarczyć treść swojego przepisu

Warunki satysfakcji :

- 1.Przepis został dodany poprawnie.
2. Przepis zostaje prawidłowo wyświetlony

Jako użytkownik chce móc dodawać zdjęcia przepisów, żeby dostarczyć elementów wizualnych do swojego przepisu

Warunki satysfakcji

1. Zdjęcie zostało dodane poprawnie.
2. Zdjęcie zostaje prawidłowo wyświetlone

Jako użytkownik chce móc oceniać przepisy, żeby zostawić feedback , dotyczący naszego przepisu

Warunki satysfakcji

1.Ocena jest widoczna na stronie

Jako użytkownik chce móc wyszukiwać przepisy, żeby znaleźć najbardziej odpowiadający na dany moment przepis

Warunki satysfakcji

1.Wyświetlają się tylko przepisy , które zawierają określone słowa , sprecyzowane w wyszukiwarce.

Jako użytkownik chce móc filtrować wyniki przepisów, żeby znaleźć przepis o określonych cechach

Warunki satysfakcji :

1.Wyświetlają się tylko przepisy , które zawierają określone słowa oraz kryteria sprecyzowane w wyszukiwarce.

Jako użytkownik chce móc obserwować innych użytkowników, po to aby dostawać powiadomienia o ich nowych przepisach

Warunki satysfakcji

1.Wyświetlenie danego użytkownika w sekcji „Obserwowani”

Jako użytkownik chce móc zalogować/zarejestrować się do serwisu, po to aby móc korzystać w pełni z funkcjonalności strony

Warunki satysfakcji

1.Poprawne zalogowanie się do serwisu(nie zakończone błędem)

2.Dostęp do wszystkich funkcji oraz swojego profilu

Jako użytkownik chce móc włączyć interaktywny stoper , aby móc mierzyć czas wykonania swojego przepisu.

Warunki satysfakcji

1.Poprawne działanie stopera

Jako użytkownik chce móc ustalać kryteria generowania jadłospisu, żeby dostosować wynik do swoich potrzeb

Warunki satysfakcji

1.Prawidłowe wyświetlenie wygenerowanego jadłospisu

Jako użytkownik chce móc opisać swój profil, aby dodać informacje o sobie

Warunki satysfakcji

1.Poprawne wyświetlenie opisu

Jako użytkownik chce móc sprawdzać ranking przepisów , aby zobaczyć co jest popularne i polecane

Warunki satysfakcji

1.Prawidłowe wyświetlenie rankingu przepisów

7.Narzedzie pracy zespołowej



W swoim projekcie wykorzystamy Trello. Jest to narzędzie do zarządzania projektami i współpracy zespołowej. Ma podobny wygląd do tablicy z przyklejanymi notatkami - projekty i zadania można uporządkować w kolumny i łatwo przenosić, aby wskazać przepływ pracy, własność projektu i status. Zadania mogą zawierać bogate informacje, w tym obrazy i załączniki do plików, a także mogą mieć przypisane terminy i inne dane śledzenia statusu. Zadania mogą również zawierać notatki, które umożliwiają członkom zespołu śledzenie postępów, a określone osoby mogą być oznaczane do działania i kontynuowania.

Trello obsługuje również integrację z setkami aplikacji i usług innych firm, takich jak dysk Google, OneDrive, czy GitHub.

8.Product backlog

| LP | Zadanie |
|----|---|
| 1 | Przygotowanie jednolitego środowiska programistycznego dla wszystkich członków projektu |
| 2. | Implementacja strony głównej (szablon-frontend) |
| 3. | Wypełnienie szablonu strony głównej sztuczną zawartością |
| 4. | Implementacja strony związanej z wyświetleniem przepisów(szablon-frontend) |
| 5. | Implementacja strony związanej ze szczegółami przepisu(szablon-frontend) |
| 6. | Wypełnienie szablonu strony ze szczegółami przepisu sztuczną zawartością |
| 7. | Implementacja strony związanej z rejestracją (szablon-frontend) |

| | |
|-----|--|
| 8. | Praca nad responsywnością 4 podstron stworzonych wcześniej |
| 9. | Implementacja strony związanej z zalogowaniem(szablon-frontend) |
| 10. | Implementacja strony związanej z rankingiem przepisu (szablon-frontend) |
| 11 | Implementacja strony związanej z ulubionymi przepisami(szablon-frontend) |
| 12 | Praca nad responsywnością 3 podstron stworzonych wcześniej |
| 13 | Implementacja interfejsu użytkownika (szablon-frontend) |
| 14. | Implementacja wyszukiwarki przepisów(szablon-frontend) |
| 15. | Praca nad responsywnością 2 podstron stworzonych wcześniej |
| 16. | Implementacja modułu z generowaniem jadłospisu |
| 17 | Projektowanie szkieletu bazy danych |
| 18 | Implementacja modułu filtrowania wyników wyszukiwania(szablon-frontend) |
| 19 | Implementacja modułu stopera (szablon-frontend) |
| 20. | Implementacja modułu komentarzy(szablon-frontend) |
| 21 | Testowanie wyświetlania zawartości oraz responsywności całego obecnego projektu. |
| 22 | Implementacja bazy danych |
| 23 | Wypełnienie bazy danych losowymi wartościami |
| 24 | Implementacja kontrolera do obsługi strony głównej |
| 25 | Implementacja kontrolera do obsługi strony z wszystkimi przepisami |
| 26 | Implementacja kontrolera do obsługi strony ze szczegółami przepisu |
| 27 | Implementacja kontrolera do obsługi strony związanej z logowaniem |
| 28 | Implementacja kontrolera do obsługi strony z związanej z rejestracją |
| 29 | Implementacja kontrolera do obsługi strony z związanej z rankingiem przepisu |
| 30 | Implementacja kontrolera do obsługi strony z ulubionymi przepisami |

| | |
|----|--|
| 31 | Implementacja kontrolera do obsługi strony z związanej z generowaniem jadłospisu |
| 32 | Testy jednostkowe systemu |
| 33 | Testy funkcjonalne |