

Projekt oraz wdrożenie reaktywnego intranetu dla firmy – wykorzystanie technologii Meteor.js, Bootstrap i MongoDB

Daniel Dereziński

Promotor: prof. dr hab. Grzegorz Wójcik

Wyższa Szkoła Przedsiębiorczości i Innowacji w Lublinie

2 lutego 2016

Celem niniejszej pracy było opracowanie oraz wdrożenie systemu intranetowego dla firmy zajmującej się produkcją oprogramowania. Jednym z podstawowych celów intranetu jest wymiana wiedzy oraz komunikacja. Zaprojektowana oraz zaprogramowana aplikacja umożliwia dodawanie artykułów, dodawania kategorii, dodawanie projektów, komunikację w obrębie projektów wraz z możliwością dodawania artykułów. Aplikacja pozwala utworzyć profil dla organizacji, zaprosić użytkowników do tak utworzonego profilu w celu podjęcia wspólnej pracy. Możliwe jest także tworzenie kont dla użytkowników nie powiązanych z żadną organizacją.

Intranet jest to sieć komputerowa ograniczająca się do komputerów na przykład w danym przedsiębiorstwie lub innej organizacji, dostępna dla jej pracowników. Intranet dostarcza szeroki zakres informacji oraz usług z wewnętrznych systemów IT organizacji, które nie są dostępne z publicznego — zewnętrznego — Internetu. Firmy Intranet dostarcza między innymi centralny punkt wewnętrznej komunikacji i współpracy. Intranet stanowi także pojedynczy punkt dostępu do wewnętrznych jak i zewnętrznych zasobów organizacji. W najprostszej formie intranet budowany jest z wykorzystaniem sieci typu *LAN* (sieć lokalna) oraz *WAN* (rozległa sieć komputerowa).

Intranet obecnie

Obecnie systemy intranetowe przekształcają się platformę do dostarczania narzędzi, są to między innymi narzędzia do współpracy pomiędzy pracownikami, zaawansowane narzędzia wymiany plików, narzędzia do zarządzania relacjami z klientami, narzędzia do zarządzania projektami czy czasem pracy. Intranet jest także platformą do wymiany wiedzy, pomysłów i idei przez co podnosi produktywność, twórczość pracowników oraz pozwala wypracowywać nowe pomysły oraz wytyczać nowe cele dla organizacji.

Intranet obecnie

Obecne systemy wydzielają także część swoich zasobów dla klientów organizacji w celu poprawy jakości oraz szybkości komunikacji pomiędzy pracownikami a klientami. Udostępnianie klientom ważnych informacji o produktach oferowanych przez organizację pozwala na odciążenie pracowników wsparcia klienta. Intranet pozwala także nie tworzenie miejsc pracy w domu poprzez udostępnianie, przez publiczny Internet, zasobów dostępnych w wewnętrznej sieci organizacji.

Aplikacja została wykonana z użyciem następujących technologii:

- **Język programowania** - JavaScript.
- **Framework** - Meteor.js.
- **Baza danych** - MongoDB.
- **Warstwa prezentacji:**
 - HTML5
 - CSS3 wraz z less
 - Bootstrap
 - AminLTE
- **System kontroli wersji** - Git

Język programowania JavaScript został użyty do zaprogramowania zarówno części serwerowej (*back-end*) jak i części odpowiedzialnej za interfejs użytkownika (*front-end*). Obecne strony WWW a w szczególności aplikacje dostępne przez przeglądarkę szeroko korzystają z JavaScript w celu dostarczenie wielofunkcyjnego oraz interaktywnego interfejsu użytkownika. Jest to skryptowy język programowania stworzony przez firmę Netscape, najczęściej stosowany na stronach internetowych. Pod koniec lat 90 XX wieku organizacja ECMA wydała na podstawie JavaScript'u standard języka skryptowego o nazwie ECMAScript.

Jest to skryptowy język programowania stworzony przez firmę Netscape, najczęściej stosowany na stronach internetowych. Pod koniec lat 90 XX wieku organizacja ECMA wydała na podstawie JavaScript'u standard języka skryptowego o nazwie ECMAScript. Najpopularniejszy obecnie sposób tworzenia stron oraz aplikacji internetowych wyróżnia trzy warstwy — warstwę struktury (HTML), warstwę prezentacji (CSS) oraz warstwę zachowania (JavaScript).

Meteor.js jest to otwarcie źródłowy framework sieciowy czasu rzeczywistego napisany w JavaScript'cie oparty o Node.js. Wykorzystuje bazę danych MongoDB, Distributed Data Protocol oraz wzorzec publikacji - subskrypcji do automatycznego rozpropagowania zmian w danych do klientów w czasie rzeczywistym. Nie wymaga od programisty pisanie jakiegokolwiek kodu synchronizacji danych. Po stronie klienta, Meteor zależy od JQuery i może być użyty z dowolną biblioteką JavaScript UI. Meteor jest rozwijany przez *Meteor Development Group*. Meteor po raz pierwszy został publicznie pokazany w grudniu 2011 pod nazwą *Skybreak*.

MongoDB to otwarty, nierelacyjny system zarządzania bazą danych napisany w języku C++. Charakteryzuje się dużą skalowalnością, wydajnością oraz brakiem ściśle zdefiniowanej struktury obsługiwanych baz danych. Zamiast tego dane składowane są jako dokumenty w stylu JSON, co umożliwia aplikacjom bardziej naturalne ich przetwarzanie, przy zachowaniu możliwości tworzenia hierarchii oraz indeksowania.

HTML5

HTML5 jest rozwinięciem języka HTML4 oraz jego XML-owej odmiany XHTML 1. Został on opracowany w ramach pracy grupy roboczej WHATWG oraz W3C. HTML5 poza dodaniem nowych elementów, usprawniających tworzenie serwisów oraz aplikacji internetowych, doprecyzowuje niejasności w specyfikacji HTML4, które przede wszystkim dotyczą sposobu obsługi błędów.

HTML5

HTML5 stawia także na semantykę. Element `<div>` traci na znaczeniu na rzecz elementów `<header>`, `<main>`, `<article>`, `<aside>`, `<footer>`, `<nav>`. Dodane zostały także następujące elementy `<canvas>`, `<figure>`, `<details>`, `<summary>`. Element `<input>` zyskał dodatkowe typy np.: *tel*, *search*, *url*, *email*, *datetime*, *date*. Dodano nowe atrybuty do elementów formularza np.: *autofocus*, *required*, *autocomplete*, *min*, *max*. HTML5 ma możliwość osadzania *MathML* i *SVG* bezpośrednio w dokumencie.

CSS3

Kaskadowe arkusze stylów — CSS to język służący do opisu formy prezentacji stron WWW. CSS został opracowany przez W3C w 1996 roku. Język ten jest potomkiem języka *DSSSL*. Pierwszy szkic specyfikacji CSS został zaproponowany w 1994 roku.

CSS3

CSS to lista dyrektyw – reguł – ustalających w jaki sposób ma zostać wyświetlona przez przeglądarkę internetową zawartość wybranego elementu (X)HTML lub XML. Można w ten sposób opisać wszystkie pojęcia odpowiedzialne za prezentację elementów dokumentu strony internetowej, takie jak rodzina czcionek, kolor tekstu, marginesy, odstępy międzywierszowe, pozycja danego elementu względem innych elementów bądź okna przeglądarki.

Less

Less (*Leaner CSS*) to dynamiczny język arkuszy stylów stworzony przez Alexis Salliera. Został stworzony w odpowiedzi na język Sass oraz dał początek nowszej wersji Sass - SCSS, która zapożyczyła część jego składni. Less było początkowo oprogramowaniem open source opartym na licencji MIT, którą zmieniono później na Apache License 2.0. Pierwsza implementacja napisana została w Ruby, później została ona zastąpiona wersją napisaną w JavaScript.

Less

Less jest zagnieżdżonym metajęzykiem – poprawny kod CSS jest również poprawnym kodem Less. Less dostarcza takie mechanizmy jak zmienne, zagnieżdżanie, mixiny, operatory oraz funkcje. Less może działać zarówno po stronie klienta, jak i serwera, jak również jego kod może być skompilowany do czystego CSS.

Bootstrap

Bootstrap jest to wolna oraz otwarcie źródłowa kolekcja narzędzi do tworzenia stron internetowych oraz aplikacji sieciowych. Zawiera szablony projektowe oparte o HTML oraz CSS dla typografii, formularzy, nawigacji, oraz innych elementów interfejsu użytkownika zawiera ona także opcjonalne rozszerzenia napisane w JavaScript. Głównym celem frameworka jest ułatwienie tworzenia aplikacji sieciowych jak i dynamicznych stron internetowych. Jest to framework front-endowy stanowi on podstawę interfejsu użytkownika.

AdminLTE

AdminLTE jest to otwarty źródłowy szablon dla aplikacji sieciowych do tworzenia paneli administracyjnych lub paneli kontrolnych. Szablon oparty jest o framework Bootstrap 3. Wykorzystuje wszystkie komponenty zawarte w frameworku Bootstrap wraz z ich wyglądem oraz przeprojektowując wygląd wielu często używanych wtyczek w celu stworzenia spójnego wyglądu interfejsu użytkownika. AdminLTE zbudowany jest w oparciu o moduły, co pozwala na jego łatwe dostosowanie do wymagań oraz rozszerzanie o nowe funkcjonalności.

Jako system kontroli wersji został użyty *Git*. Git jest to rozproszony system kontroli wersji. Został stworzony przez Linusa Torvaldsa jako narzędzie wspierające rozwój jądra Linux. Git jest to wolne oprogramowanie i został opublikowany na licencji GNU GPL w wersji 2.

Najważniejsze założenia aplikacji Intranet:

- tworzenie oraz rejestracja organizacji;
- tworzenie oraz rejestracja użytkowników;
- dodawanie utworzonego użytkownika do organizacji;
- organizacja w swoim zakresie, powinna posiadać główny zbiór artykułów — główne wiki
- główne wiki ma posiadać główną kategorię, której nie można usunąć;
- tworzenie, edycja, usuwanie kategorii artykułów w głównej wiki;
- tworzenie, edycja, usuwanie artykułów w głównej wiki;
- artykuły mogą być dodawana do głównej kategorii wiki lub do utworzonych przez użytkownika kategorii;
- edycja oraz prezentacja artykułów na obsługiwać język znaczników *markdown* w szczególności jego implementację *GitHub Flavored Markdown* — GFM;

- organizacja oraz użytkownicy mogą tworzyć, usuwać projekty w swoich zakresach lub zakresach, do których należą;
- projekty powinny posiadać stronę z podsumowaniem;
- projekty powinny posiadać niezależne wiki;
- projekty powinny mieć możliwością dodawania — zapraszanie — użytkowników oraz ich usuwanie;
- wiki dla projektów ma mieć te same cechy jak główne wiki;
- projekty powinny posiadać możliwość komunikacji pomiędzy użytkownikami przynależącymi do danego projektu;
- komunikacja w obrębie projektu mam mieć możliwość dodania wiadomości wraz z jej tytułem oraz możliwością dodania komentarzy;

Niezależność od platformy oraz dostępność na jak największej ilości urządzeń narzuciła wybór typu aplikacji. Budowana aplikacja będzie typu sieciowego. Klienci będą uzyskiwać do niej dostęp poprzez przeglądarki internetowe kontaktując się z serwerem, na którym będzie dostępna budowana aplikacja.

W oparciu o założenia jakie ma spełniać aplikacja *Intranet* oraz o obecne trendy w rozwoju aplikacji sieciowych do budowy aplikacji został wybrany framework Meteor.js, a jako język programowania JavaScript. Wybór tych technologii był podyktowany także szybkością oraz prostotą budowania aplikacji. Meteor.js używa JavaScriptu zarówno po stronie klienta oraz serwera co skraca czas nauki samego framework'a eliminując potrzebę nauki dodatkowego języka programowania. Tematyka oraz założenia projektu nie stanowią problemu dla przyjętego przez framework Meteor.js nierelacyjnego rozwiązania bazodanowego.

Aplikacja ma być dostępna na wszystkich znaczących platformach oraz jak największej ilości urządzeń. Aby to zapewnić w warstwie prezentacji wykorzystano HTML5, CSS3 wraz z kompilatorem less oraz domyślnie wykorzystywaną przez Meteor.js bibliotekę JQuery. W celu zachowania responsywności interfejsu użytkownika wykorzystano szablon AdminLTE, który oparty jest o framework Bootstrap. Wykorzystanie Bootstrapa dostarcza prosty oraz szybki sposób na dostosowanie układu i wyglądu interfejsu użytkownika w zależności od używanego przez niego urządzenia. Rozwiązanie to znaczenie zwiększa użyteczność aplikacji podczas używania urządzeń o różnych wielkościach oraz rozdzielczościach ekranu.

Aplikacje napisane z użyciem frameworka Meteor.js mają dość specyficzne wymagania co do serwera na jakim będą uruchamiane. Meteor.js oferuje swoją własną implementację MongoDB. Jednak do celów produkcyjnych nie jest wskazane wykorzystanie tego rozwiązania. Z tego powodu hosting musi posiadać normalną wersję bazy MongoDB. Jedynym gotowym takim rozwiązaniem jest hosting oferowany przez twórców Meteor.js.

Druga opcja to serwer współdzielony, bądź dedykowany, na którym mielibyśmy pełną kontrolę. Jako system operacyjny najlepiej oraz najtaniej jest wykorzystać, którąś z popularnych dystrybucji Linu-xa. Najlepsze efekty można uzyskać wykorzystując osobne serwery na aplikację oraz bazę danych. Do tego celu można wykorzystać wirtualizację z wykorzystaniem Dockera, opartego na tak zwanych kontenerach (*linux containers*).

Aktualnie aplikacja jest w fazie testowania przez pracowników firmy. Naprawiane są zgłaszane błędy. Aplikacji brakuje także kilku kluczowych elementów jak na przykład zarządzanie użytkownikami w organizacji, panel do administrowania całością aplikacji, ustawienia użytkowników. Brakuje także dokumentacji, która jest niezbędna do poprawnego wdrożenia aplikacji.

Powyższa praca miała na celu zbudowanie oraz wdrożenie reaktywnego intranetu dla firmy z sektora IT. Aplikacja powstała z wykorzystaniem najnowszych trendów oraz technologii w dziedzinie wytwarzania aplikacji sieciowych. Nowoczesne aplikacje sieciowe coraz bardziej upodobniają się do ich biurowych odpowiedników. Głównie dotyczy to interfejsu użytkownika oraz sposobu jego działania, który to na przestrzeni kilku ostatnich lata podlegał dużym zmianom.

Opisane w pracy oraz pokazane w postaci aplikacji techniki, technologie oraz rozwiązania spełniają wszystkie wymagania jakie stawia się dziś nowoczesnej aplikacji sieciowej. Jedynym wyjątkiem jest komunikacja z serwerem. Wykorzystując framework Meteor.js w miejsce np REST'owego API dostajemy coś bardziej nowoczesnego oraz o wiele ciekawszego.

Opierając się na DDP (*Distributed Data Protocol*) zyskujemy mniej pracy oraz czasochłonny sposób na synchronizację danych pomiędzy serwerem a klientem, który w działaniu jest bardzo spektakularny. Nie musimy pisać kodu synchronizującego dane, framework sam to realizuje. Meteor.js wpisuje się w trend używania jednego języka programowania po stronie serwera jak i klienta.

Wdrożenie aplikacji jest w wstępnej fazie. Aplikacja podlega testom wykonywanym przez pracowników firmy. Po usunięciu błędów, dodaniu brakujących elementów aplikacji oraz po uzupełnieniu dokumentacji aplikacja zostanie wdrożona z wykorzystaniem serwera współdzielonego lub dedykowanego.