**WYŻSZA SZKOŁA**

**INFORMATYKI i ZARZĄDZANIA**

**z siedzibą w Rzeszowie**

**WYDZIAŁ INFORMATYKI STOSOWANEJ**

**Kierunek: INFORMATYKA**

**Specjalność: Programowanie**

Oleksandr Lisivka

Nr albumu 53410

***Serwis oglószeń do podroży***

Promotor: dr Marek Jaszuk

**PRACA DYPLOMOWA INŻYNIERSKA**

**Rzeszów 2018**

Ja niżej podpisany/a oświadczam, że składana przeze mnie praca dyplomowa pt. „...........................” została przygotowana samodzielnie.

Oświadczam również, że praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna ze złożoną wersją elektroniczną.

.........................................................

data czytelny podpis autora

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

................................................................

data czytelny podpis promotora

Оглавление

[**Wstęp** 6](#_Toc508558901)

[**Uzasadnienie wyboru tematu pracy** 6](#_Toc508558902)

[**Cel pracy** 7](#_Toc508558903)

[**Aktualny stan wiedzy dotyczący problemu (zagadnienia) będącego przedmiotem pracy** 7](#_Toc508558904)

[**Zakres pracy** 7](#_Toc508558905)

[**Omówienie układu pracy, krótka charakterystyka poszczególnych rozdziałów** 8](#_Toc508558906)

[**Część główna pracy (podzielona na numerowane rozdziały, ewentualnie podrozdziały).** 9](#_Toc508558907)

[**Część teoretyczna** 9](#_Toc508558908)

[**Część praktyczna** 9](#_Toc508558909)

1. Strona tytułowa pracy (pierwsza strona pracy).
2. Czytelnie podpisane oświadczenia: autora oraz promotora pracy (druga strona pracy).
3. Spis treści (trzecia strona pracy).
4. Wstęp, w tym cel pracy.
   1. uzasadnienie wyboru problematyki/tematu pracy,
   2. cel pracy
   3. aktualny stan wiedzy dotyczący problemu (zagadnienia) będącego przedmiotem pracy,
   4. zakres pracy,
   5. omówienie układu pracy
5. Część główna pracy (podzielona na numerowane rozdziały, ewentualnie podrozdziały).
   1. Część teoretyczna
      1. Front-end
         1. Html5
         2. CSS3
            1. SCSS
            2. Bootstrap
         3. JavaScript
            1. JQuery
      2. Back-end
         1. Node.js
         2. Npm
            1. body-parser
            2. ejs
            3. express
            4. express-session
            5. method-override
            6. mongoose
            7. nodemon
            8. passport
            9. passport-local
            10. passport-local-mongoose
         3. Baza danych
            1. Table relations
            2. NONSQL database

Mongoose

* + - 1. REST architectura
         1. REST routing
      2. Git
         1. GitHub
      3. IDE
         1. Microsoft Visual Studio Code
  1. Część praktyczna
     1. Specyfikacja wymagań
        1. Wymagania funkcjonalne
        2. Wymagania niefunkcjonalne
        3. Diagram przypadków użycia
        4. Mockup
     2. Git repozytorium
     3. Tworzęnie aplikacji
        1. Front-end
           1. JS
           2. CSS
        2. Back-end
           1. Routing
           2. Baza danych

Shema

* + - * 1. Autoryzacja
        2. Authentyfikacja
    1. Sprawdzenie działania aplikacji

1. Zakończenie.
2. Literatura.
3. Streszczenie pracy.
4. Załączniki.

**Wstęp**

W raz z wynalezieniem internetu wsiat się zmiania i się rozwija co raz szybciej. Cały świat jest powiązany z każdą jego częścią za pomocą połączenia internetowego. Jednosześnie z tym w każdym momencie mająć dostęp do sięci również masz dostemp do wszystkich jego serwisów ułatwiających życie. Komunikacja od czasów jej wynależenia w życiu codziennym odgrywa najważniejszą role polączająć ludzi. Do wynależenia internetu ludzie komunikowały się najpierw ustnie póżniej pisemnie i już mogły przekazywać istotną dla nich informację na odłegłość. Progres ze swojej koleji w metodach przekazywania informacji wystąpił odnośnie nie tak dawno licząć od początkó istorii. I teraz mająć za sobą podgruncie komunikacja ma możliwość lączyć nie tylko osobnych ludzi ale i kompanie i nawet urądzenia (Internet of things). Praca i byt codzinny coraz większej ilości osób i przedsięmbiorstw jest związany za pomocą tego cuda - globalnego systemu połączeń sieciowych. Jednocześnie z tym ilość uzależnionych od internetu też co raz rośnie ale to nie jest niczym dziwnym kiedy idziesz w nogę z czasem. Komputery, laptopy, telefony, zegarki, stacji gier, konsole i wiele innych rzeczy komunikują się z innymi rzeczami czy nawet serwarami i ilość danych codzięnnie rośnie.

W dniach dzisięjszych żeby dostać to czego chcesz nie potrzebujesz dużo. Również jak i komunikowanie sprawa wymiany czy póżniej sprzedaży rzeczy lub usługi pojawiła się bardzo dawno a teraz ma swoj kolejny pik. Teraz żeby kupić rzecz nie koniecznie musisz iść gdzieś i jej szukać, wybierać czy porównywać. W ostatnich latach się rozwija sprzedaż za pomocą internetu. Sklepy, serwisy internetowe, salony, fabryki i temy podobne rzeczy można odnaleść na stronach WWW(World Wide Web)(1).Co raz więcej towarów i usług dostępne w postaci elektronicznej. Dość prosty i zrozumiały z ekonomii wyraz: popyt powoduje propozycję. Tak i z serwisami internetowymi. Popularyzacja serwisów internetowych powoduję powiększenie ich ilości. Prosty i zrozumialy wygląd tranzakcji znajomy wszystkim ze swiatu rzeczywistego został przeniesiony do internetu. Teraz każdy użytkownik może występować jako sprzedawca czy jako konsument. Ze swojej strony to powoduję przyrost serwisów w każdej branży i dziedzinie.

Dla przykładu żeby teraz dostać się z punktu A do punktu B wystarczy pobrać czy otworzyć mapę i jechać za marszrutem z listy proponowanych. Albo wyszukać i wybierać samochód co jedzie za potrzebnym tobie kierunkiem i umawiać się z kierowcem żeby oczekiwał że pojedziesz z nim, kiedy i gdzie jest z min spotkanie. Jeździć że przy pomocy koleji też nie jest tak trudno. Dane o przyjazdach, wyjazdach i cenach są dostępne każdemu i w każdej chwili w internecie. Loty samolotem też są podobne. Wpisująć miasto wylotu oraz docelowe zostaniesz proinformowany o rejsach i kierónkach za jakimi oni lecą. Kupujesz bilet i jesteś gotowy do podróży – wsiadasz i lecisz. Na podwtawie tych danych teraz dzialają biura podróży. Mająć dostemp do danych o wylotach czy odjazdach i przyjazdach busów czy innych środków transportu wybierają skąd i dokąd i rezerwują hotel dla wybranej osoby na wybrany okres czasu. Czyli podają informacje hotelowi który może się znajdować na przeciwnej stronie kuli zięmnej. I oni mająć taki komunikat już oczekują że w czasie rezerwacji przyjedzie wybrana ilość osób.

Są również serwisy na których można zarezerwować miszkanie w określonym terminie albo orendować pokój. Jest to dość prostym i użytecznym kiedy planujesz sobie wyjazd do innego miasta. Takie serwisy są dobrą zamianą rezerwacji hotelów kiedy nie planujesz albo nie masz możliwości wydawać dużo za nocleg w innym miaście czy panstwie. Ale czy są serwysy pozwalające do rezerwacji kulturnej programy wycieczki. Jęzeli porównywać z hotelami to są serwisy gdzie być mógł to zrobić ale na przykład jako dodatek od biura podroży. Serwis na kórym można by było samemu powiedzieć czego szukasz i zaplanować to wspolnie z kimś albo dołonczyć się do kogoś w już zaplanowanej podróży, czy zarezerwować wobie wycieczka po mieście u tego kto ją proponję jeszcze niema.

**Uzasadnienie wyboru tematu pracy**

Potrebności na rynku rosną co raz szybciej. Dowodem tego jest każdy popujarny produkt. Czasami żeby dostać produkt jednym z pierwszych ludzie stoją w kolelce dniami i noczami. Żeby nie przegapić moment jakiegoś zapotrzebowania na rynku potrzebno cały czas myślić na przód. Serwis dla podrożujących gdzie użytkownicy by moli znaleść kogoś do wspolnego wyjazdu czy żeby znalść przewodnika po mieście do którego jadą był by bardzo przydatny podrożującym. W drugą stronę ludzi by moli sami dodawać ogłoszenia i proponować innym się do nich dołonczyć. W danym momencie rynek jeszcze nie ma podobnego produktu w swojej branży, temu on by był wybrany jako temat mojej pracy dyplomowej.

**Cel pracy**

Celem pracy było by stwożyć serwis dla pomocy podrożującym na którym oni by mogli dodawać ogłoszenia o swoich planach na wyjazd do którego potrzebna jest większa kompania lub dodania komentaży w odpowiedź do danego ogłoszenia.

Praca w swoją kolej była by podzielona na stwożenie stron internetowych do danego serwisu oraz web serwisu po stronie którego wykonywałybysię działania na stronach. Serwis powinienen zawierać możliwości logowania, dodowania ogłoszeń, edytowania ogłoszeń i ich usuwania przeż osobe co ich dodała. Komentaże równiż piwinni mieć podobną funkcjonalność. Strony internetowe powinne być responsywne i mieć prostą strukture żeby użytkownicy się w niej nie gubili oraz żeby strona nie była przepęłniona kontentem który był by trudnym w ogarnięciu (nie był by nieżbędnym)dla wykonania konkretnego działania na stronie(User Experience).

**Aktualny stan wiedzy dotyczący problemu (zagadnienia) będącego przedmiotem pracy**

Przed rozpocięciem danej pracy ja(Oleksand Lisivka) miałem doczynienie ze stronami internetowymi podczas praktyk zawodowych od uczelni. Pod czas danych praktyk nauczyłem się i pogłembiłem poznania w części Front-end’owej, czyli z HTML5, CSS(SCSS, SASS, Bootstrap), JavaScript(jQuery). Odnośnie wiedzy niezbędnej do napisania części serwerowej w wybranym przezemnie języky(JavaScript w środowisku Node.js), to takiej wiedzy nie posiadałęm. Na uczelni nauczyłem się języka C# i wzorzeca architektoniczny służącego do organizowania struktury aplikacji MVC(2) i mółbym wykonać pracę na podstawie danej więdzy ale wybrałem inny język żeby ciągle uczyć się czegoś nowegoktóry i aby się rozwijać ponieważ w dziedzinie Technologij informacyjnych języki, techniki, standarty, i technologie ciągłe się zmieniają i potrzebno iść razem z nimi. Typ dazy danych (NONSQL) był wybrany z tego samego powodu - żeby powiększyć więdze profesjonalną.

**Zakres pracy**

Cały system można podzielić na części które muszą być zrealizowane w kolejności w której muszą być dodawane do serwisu:

* REST Routing
* Tworzęnie stron
* Twożenie ogłoszenia
* Konfigurowanie bazy danych i podpinanie jej do aplikacji żeby dać możliwość zapisywania danych do niej
* Dodawanie komentaży
* Logowanie
* Autentyfikacja
* Edytowanie lub usunięcie ogłoszenia przez go wsaściciela
* Edytowanie lub usunięcie komentaży przez ich wsaścicieli
* Wyświetlenie komunikatów dla użytkownika
* Responsywność stron do różnych rozszeżeń ekranu.
* Podtzymywanie różnych wersii przeglądarek internetowych.

Dana lista swkazuje na kolejność realiowanych zadań ze wzgłedu na ich ważność dla realizacji danego projektu. Idąć z góry do dołu tworzymy system od śkieletu do gotowego produktu.

Na przykład żeby każdy użytkownik mógł by utworzyć ogłoszenie on musi najpierw przejść na odpowiednią strone i się zalogować. Z punktu widzenia serwisu bez logowania dla wykonania rządania do serwera z informacją o ogłoszeniu iżytkownik musiał by przejść się przez 4 pierwsz części z listy, czyli otwożyć strone pod potrzebnym adresem(REST Routing). Po tym użytkownikowi musi wyświetlić się strona z treścią która odpowiada adresowi(Tworzęnie stron). Uzytkownik tworzy odpowiedznie rządanie do serwera z treścią ogłoszenia i po stronie serwera po przetwarzaniu danych musi się utworzyć ogłoszenie(Twożenie ogłoszenia oraz Konfigurowanie bazy danych i podpinanie jej do aplikacji żeby dać możliwość zapisywania danych do niej). Dalej dla rozszeżenia funcjanalności dodajemy po stronie serwera odpowiednie ścieżki routingu i strony co by się renderowały po przejściu na nich dla dodawania komentaży i ścieżki do oczekujące na wchadzące od użytkownika dane o komentazu(Dodawanie komentaży). Więcej szczegulowo informacja zostanie podana w kolejnych rozdziałach pracy.

**Omówienie układu pracy, krótka charakterystyka poszczególnych rozdziałów**

Praca dla latwości jej przetważania zostanie podzielona za treścią na części (np. część wstępu, część główna pracy, zakończenie, literatura, streszczenie pracy, załączniki.). Wstęp oraz część glówna zostaną podzielone na działy(np w przypadku części głuwnej na część teoretyczą oraz część praktyczną). Żeby działy ze swojej koleji zostały się więcej formatowanymi żeby odrazu więdzieć do częgo należy część pracy w uśczegulonych poddziałach została by podana śczegulniejsza informacja(np. część frontendowa część backendowa).

Charakterystyka nie potrzebuję śczegulnego omusienia z tego powodu że o treści każdej z części było by wiadomo odrazu z miejsca w którym ona się znajduje(np. Część teoretyczna -> Front-end -> Html5 miała by w sobię teorije o HTML5 co się odnosi do części frontendowej).

**Część główna**

Dany rozdział bezpośrednio odpowiada za opisanie nażędzi wykorzystanych przy tworzeniu danej pracy oraz za przebieg pracy pod czas jej realizacji od początku do konca. Z tego powodu została ona podzielona na 2 działy różnące się pomiędzy sobą.

**Część teoretyczna**

Poniżej są opisane tehnologie oraz narzędzia które zostały użyte przy napisaniu danej pracy dyplomowej. Niektóre są wymagane dla jej realizacji(np. HTML5 jest teraz standartem w napisaniu stron internetowych i pisać strone dla przykładu na tabelach nie było by dotrzymaniem się standartów) a niektóre zostały wybrane ze wzgłendów na łatwość w użyciu oraz możliwości które daję danę nażedzie, ciągle wsparcie i rozwijająnie(np. Bootstrap, jQuery czy npm).

Dwie najważniejsze części projektu to oczywiście front-end i back-end(3). Front-end odpowieda za komunikowanie się z użytkownikiem, pobieranie danych od użytkownika, czy ich wyświetlenie oraz przekazanie ich do back-endu. Back-end jest mózgiem i sercem aplikacji ponieważ w nim na podstawie danych wykonują się określone zadania. To nie znaczy że front-end jest mniej ważny. Dużo czego można zrobić za pomocą JavaScripu ale żeby nikt nie mał na to wpływu i żeby to było chroniono od innych żeczy ważniejszę są zaimplementowane po stronie serwera. Front-end w tej części występuję jak pyszy deser go głównego dania czym jest część serwerowa, jako graficzna reprezentacja dla aplikacji konsolowej. Taki typ podzielenia obowiązków set dość typowym i popularnym pośród oprogramowania danego typu. Tak podzielona praca daje osobie tworzącej strone wolne rzęce i jednocześnie dataa osoba nie koniecznie musi znać cech serwera i co się na nim dzieje, a programista serwera nie potrzebuje znajomości HTML i CSS bo zajmuję się nie wyglądem a tylko funkcjonalnością.

**Front-end**

Fornt-end jest częścią aplikacji która odpowiada za komunikację z użytkownikiem przy pomocy interfejsu graficznego. Dość często front-end jest używany jako pośrednik pomiędzy użytkownikiem systemu a aplikacją na serwererze. W systemach takiego typu front-end posługuje dla wyświetlenia informacji oraz pobierania informacji od użytkownika. Dość proste strony czy np. emaili mogą nie mieć serwera czy w przypadku emaili nawet scryptów. Takie strony wykorzystywane do reprezentacji danych. Są również i typy stron pomiędzy dwoma powyższymy. Takie strony nie są zbyt skomplikowane żeby niezbędny był serwer ale i nie są statyczne. Takie strony napisane są przy pomocy języka JavaScript dla dodawania interaktywności na strony. Więczej szczęgulowo każdy z elementów jest opisany poniżej.

**HTML5**

HTML albo ang. HyperText Markup Language(4) jest językiem znaczników do markowania, strukturyzacji i prezentowania treści w Internecie. Cyfra 5 w nazwie wskazuję na aktualną wersję – piątą, dlatego wykorzystanie języku ze standartami ostatniej wersji zostało nazwane HTML5 co połączyło w sobie normy i standarty HTML oraz XHTML. HTML był stworzony żeby opisać strukturę informacji zawartych wewnątrz strony internetowej w postaci modelu oiektowego(DOM(5)). HTML z tekstu który ma buduję hiperłącza, akapity, nagłówki i temu podobne rzeczy. Bez dodawania CSS strona z otwierających i zamykających się tegów ma swoją strukture ale nie ma żadnego wyglądu. Jest to „ściana tekstu” dla maszyny żeby ona wiedziała kiedy jedna go część się kończy a kolejna się zaczyna.

Struktura modelu DOM (ang. Document Object Model) przedstawia dokument HTML w postaci struktóry drzewa. Model DOM definiuje obiekty i właściwości wszystkich elementów HTML oraz metody dostępu do nich. Korzystając się z DOMu przy pomocy JavaScriptu można łatwo dodać zmienić lub usunąc element.HTML.

**CSS3**

Kaskadowe arkusze stylów(Cascading Style Sheets)(6) jest językiem służącym do opisywania wyglądu, czyli wyświetlenia strony internetowej. CSS jest listą regul które wskazują przeglądarce w jaki sposób dana strona musi być wyświetlona i jak wyglądają wybrane elementy. CSS jest używany dla określenia kolorów, czcionek i innych aspektów wizualizacji stron internetowych. Był on stworzony żeby trzymać osobno strukture dokumentu a jego stylowanie. Z tego powodu strony na których stylowanie jest wpisane liniowo do kodu HTML musą być dawno napisane i tego się nie zmienia żeby trzymać kod jednolitnym. Czytać takie strony staje drudno kiedy przykladowo na jedą linię HTML przychodzi się po 3 stylowania. Również użycie klas pomogło by trzymać kod suchym(7) i nie powtarzającym się. Osobny plik ze stylowaniem może być użyty na różnych stronach co ułatwia o wiele ich napisanie.

**SCSS**

FILL ME

**Bootstrap**

Bootstrap(8,9) jest frameworkiem(10) dla CSS, stworony przez programistów Twittera. Glówną cechą bootstrapu jest responsywność oraz już gotowe klasy które w latwy i szybki sposób pomagają przestylować stronę internetową. Bootstrap napisany jest w taki sposób żeby strona również dobrze wygłądała w większościi najpopularniejszych przeglądarek, takich jak Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera i Safari. Bótstrap jest łatwy w użyciu i pomaga zaośczędziś dużo czasu ns stulowaniu stron. Dla działania danego frameworku niezbędnym jest wykorzystanie biblioteki jQuery. Dodając trzy referencje do swojego kodu (pierwsza do stylowania, druga do jQuery i trzecia do scryptów Bootstrapowych) do pobranych plików lub z CDN serwisów można używać go klas w każdej chwili.

**JavaScript**

JavaScript(11) jest wieloplatformowym objekto orijentowanym językiem scryptowym i jest trzecim ale nie za ważnością tytanem front-endu. JavaScript jest bardzo wielofunkcojalnym poniważ jest wykorzystywany w przeglądarkach, po stronie serwera(NodeJS), mobilnych ta desktopowych aplikacjach. W przeglądarkach JavaScript występuję jak na stronach tak i w web rozrzerzeniach. Strona bez wykorzystania JS wygląda statychną i nie zmieniającą się ale z dodawaniem JS wzsystko się zmienia. Dla przykładu JS jest używany do stworzenia scenariuszy za którymi strona się zachowóję co daje możliwość na stronie współdziałać z końcowym użytkownikiem na go użądzeniu. Również dobrze i często wykorzystywany do walidowania formólarzy na stronie, wymieniać się danymi z serwerem, zmieniać struktóre ta widoki stron.

Czasami przy stylowaniu strony potrzebno wybrać jakieś elementy ze strony ale stylowanie przez CSS nie daje takich możliwości do przestylowania ich i JS jest jedynym wyjściem. Z drugiej strony tak jak HTML CSS oraz JavaScript są różne za swoim znaczeniem warto by przymać ich osobno o ile się to daję. Niestety nie wszystkie przegliądarki dzialają jednakowo i temu jak i niektóre stylowania tak i niektóre scrypty mogą nie działać wszędzie, temu przy napisaniu kodu warto uważać na przęgądarki i ich werję. Większość problem można rozwiązać prototypowaniem (JS jest językiem z koncepcją prototypów temu podtzymowanie objektów jest takie samo i się nie różni od tradyctjnych języków OOP(12)) Takie problemy są z czasem rozwiązywanę po przez standartyzację.

Standartem ostatniej wersji JS jest ECMAScript wersji 6. ECMAScript jest ustandaryzowany przez ECMA obiektowy skryptowy język programowania. Specyfikacja go znana jest jak ECMA-262. ECMA-262 definiuje semantykę języka ECMAScrpt(ktrego implementacją jest JavaScript oraz innę), oraz np. podstawowe typy danych (String, Boolean, Number, Object itd.) i obiekty (np. Math, Array).

**JQuery**

JQuery(13,14) jest szybką, niewięlką i bardzo popularną biblioteką języka JavaScrip. Powiedzieć popularna przy okraśleniu może nie wystarczyć ponieważ jQuery jest najpopujarniejszą biblioteką języka JavaScript i wykorzystuję się na ponad polowie najpopularniejszych stron internetowych. Pomaga ona w wybieraniu objektów DOMu oraz go manipulacją. Animajca, tworzenie eventów i ich obsługa jest też o wiele latwiejsza z jQuery. Ilość kodu napisanego z jQuery jest mniejszą, temu i latwiejszą w go czytaniu. Nie koniecznie zawsze potrzedno używać jQuery ale licenzja MIT(15) pozwala każdemu używać danej biblioteki za darmo w każdej chwili.

**Back-end**

Back-end na przeciwnictwo front-endowi nie działa po to żeby kontaktować się z użytkownikiam tylko po to żeby generosać kod maszynowy dla procesorów z różnymy przyznaczeniami. Front-end wytępuje pośrednikiem pomiędzy back-endem a użytkownikiem koncowym, temu programista części fronentedowej jest odpowiedzialny za wyróżnienie aplikacji wykonanej przez programistę części backendowej. Dla przykładu mająć do porównania dwa serwisy internetowe dla sprzedaży back-endowe części mogły by mieć wiecej wspólnego pomiędzy sobą w kodie źródłowym aniż frontendowe. Oczywiście back-end mógł by być napisany w różnych językach i z użyciem różnych platform ale funkcjonalność napisana po stronie back-endu miała by większe podobięstwo aniż wygląd po stronie front-endu. Tak jak front-end dostępny każdeny użytkownikowi jakiejś strony to back-end odpowiadający za działalność całej aplikacji, i jej funkcjonal jest ukryty od większości. Przyczyn na to jest wiele od tego że korzystająć się ze strony nie potrzebujesz wiedzieć co się dzieje „pod spodem” takiej strony żeby nadał normalnie jej używać kończąć polityką bezpieczęśtwa. Końcowy użytkownik nawet nie zawsze może wiedzieć że coś się zmieniło po stronie serwera, dla przykładu kiedy w 2011 Twitter zmienieł swają część back-endową z Ruby on Rails na Java końcowe użytkownicy mająć dotyczenie tylko z front-endem nawet nie zauważyli różnicy.

Programowanie back-endowe to programowanie, którego celem jest implementacja strony serwera integracja bazy danych i połączenie jej ze stroną użytkownika(front-end). Mówiąc najprościej, frontend przesyła informacje i polecenia od użytkownika do serwera, który z kolei je przetwarza.

**Node.js**

Node.js(16, 17) jest to środowisko zaprojektowanedo twożenia aplikacij internetowych napisanych w języku Javascript

**Npm**

asasd

* + - * 1. body-parser
        2. ejs
        3. express
        4. express-session
        5. method-override
        6. mongoose
        7. nodemon
        8. passport
        9. passport-local
        10. passport-local-mongoose
      1. REST architectura
         1. REST routing
    1. Baza danych
       1. Table relations
       2. NONSQL database
          1. Mongoose
    2. Git
       1. GitHub
    3. IDE
       1. Microsoft Visual Studio Code

**Część praktyczna**

* + 1. Specyfikacja wymagań
       1. Wymagania funkcjonalne
       2. Wymagania niefunkcjonalne
       3. Diagram przypadków użycia
       4. Mockup
    2. Git repozytorium
    3. Tworzęnie aplikacji
       1. Front-end
          1. JS
          2. CSS
       2. Back-end
          1. Routing
          2. Baza danych

Shema

* + - * 1. Autoryzacja
        2. Authentyfikacja
    1. Sprawdzenie działania aplikacji

Zakończenie

Literatura

1. <https://en.wikipedia.org/wiki/World_Wide_Web>
2. <https://en.wikipedia.org/wiki/Model–view–controller>
3. <https://en.wikipedia.org/wiki/Front_and_back_ends>
4. <https://en.wikipedia.org/wiki/HTML>
5. <https://en.wikipedia.org/wiki/Document_Object_Model>
6. <https://en.wikipedia.org/wiki/Cascading_Style_Sheets>
7. <https://en.wikipedia.org/wiki/Don't_repeat_yourself>
8. <https://getbootstrap.com/>
9. <https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)>
10. <https://en.wikipedia.org/wiki/Software_framework>
11. <https://en.wikipedia.org/wiki/JavaScript>
12. <https://en.wikipedia.org/wiki/Object-oriented_programming>
13. <https://jquery.com/>
14. <https://en.wikipedia.org/wiki/JQuery>
15. <https://en.wikipedia.org/wiki/MIT_License>
16. <https://nodejs.org/en/>
17. <https://en.wikipedia.org/wiki/Node.js>
18. <https://en.wikipedia.org/wiki/URL>

**Zalecana zawartość treści pracy**

Wstęp

Kilkustronicowa analiza ogólna wprowadzająca w realizowane zagadnienie, zarysowująca problematykę pracy i użyteczność wynikającą z jej realizacji (uzasadnienie realizowanej problematyki). W ramach wstępu należy wyszczególnić następujące elementy:

* uzasadnienie wyboru problematyki/tematu pracy,
* cel pracy
* aktualny stan wiedzy dotyczący problemu (zagadnienia) będącego przedmiotem pracy,
* zakres pracy,
* omówienie układu pracy, to jest krótka charakterystyka poszczególnych rozdziałów.

Pod pojęciem celu pracy dyplomowej na studiach pierwszego stopnia należy rozumieć nie tyle samo rozwiązanie problemu, ile korzyści jakie ono przyniesie (cel to przyszły efekt podjętego działania - rozwiązywania problemu - który działanie to uzasadnia). Cel pracy powinien być tak sformułowany, aby możliwa była ocena efektywności działań podjętych w ramach jego rozwiązywania oraz stopnia osiągnięcia założonego celu.

Część główna pracy (podzielona na numerowane rozdziały, ewentualnie podrozdziały)[[1]](#footnote-1)

Część główna pracy powinna być dostosowana do specyfiki i przedmiotu rozwiązywanego problemu (zagadnienia) i składać się z części teoretycznej i praktycznej.

W części teoretycznej autor powinien dokonać charakterystyki problematyki pracy, analizy aktualnego stanu rzeczy z określeniem pojawiającego się problemu oraz wskazać potencjalne kierunki jego rozwiązania z uzasadnieniem wyboru jednego z nich.

W części praktycznej autor powinien wykazać się znajomością adekwatnych do kierunku studiów procedur (metodologii) rozwiązywania problemów, umiejętnością określania stopnia osiągniecia założonego celu pracy oraz wyciągania na tej podstawie wniosków na temat uzyskanych wyników i efektywności podjętych działań.

Zakończenie

Zakończenie powinno zawierać ocenę skuteczności podjętych działań i uzyskanych efektów pod kątem stopnia osiągniecia założonego celu pracy.

Literatura

Należy podać wszystkie źródła, które wykorzystano w pracy, w tym również - oprócz aktów prawnych, wydawnictw zwartych i ciągłych - adresy stron www.

**I. ZASADY FORMATOWANIA TEKSTU PRACY**

* Plik z pracą dyplomową należy zapisać jako dokument programu Word 97-2003 (\*.doc).
* Nie należy wstawiać nagłówków i stopek z oddzielną treścią, np. tytułem rozdziału.
* Każdą część pracy należy rozpoczynać od nowej strony.
* Nie należy stawiać kropki po tytule danej części pracy (w tym rozdziału/podrozdziału) oraz po tytule pracy.
* Numeracja stron - obowiązkowo automatyczna (w stopce, wyrównana do środka strony, Czcionka Times New Roman 12 pkt.). Pierwszą stroną pracy jest strona tytułowa i od niej należy zacząć numerowanie, ale nie umieszczać na niej numeru strony.
* Automatyczny spis treści – na trzeciej stronie pracy dyplomowej.
* Czcionka - Times New Roman, rozmiar 12 pkt., grubość normalna.
* Odstępy między znakami – standardowe (0 pkt.).
* Pomiędzy wyrazami nie należy wstawiać więcej niż jednego znaku spacji.
* Odstępy między wierszami - pojedyncze.
* Tekst justowany obustronnie.
* Włączona opcja automatycznego dzielenia wyrazów.
* Jeżeli dany akapit ma znaleźć się na nowej stronie należy zastosować funkcję podziału strony (kombinacja klawiszy: *Ctrl + Enter*).
* Na końcu wiersza nie mogą występować pojedyncze litery lub spójniki. W celu przeniesienia spójnika do kolejnego wiersza należy użyć tzw. twardej spacji (kombinacja klawiszy: *Ctrl* + *Shift* + *spacja*). „Twardą spację” należy wstawić bezpośrednio po spójniku, a przed następującym po nim wyrazem (powoduje to „przyklejenie” spójnika do wyrazu).
* Zabrania się wstawiania na końcu wiersza (w środku zdania) zwykłych oraz „miękkich” enterów.
* Pomiędzy wyrazem a następującym po nim znaku przestankowym (przecinek, kropka, średnik, dwukropek) nie należy wstawiać znaku spacji.
* Wyrazy (zdania) pisane w nawiasie nie mogą być oddzielone od tych nawiasów spacjami.
* Wyliczenia w tekście - za pomocą automatycznego wypunktowania lub numerowania.
* Praca musi być wydrukowana dwustronnie - w zakładce *Plik/Ustawienia strony/Marginesy* należy zaznaczyć „marginesy lustrzane” (margines wewnętrzny - 2 cm, margines zewnętrzny, górny i dolny - 1,5 cm).

**II. STRONA TYTUŁOWA PRACY DYPLOMOWEJ**

Strona tytułowa pracy dyplomowej powinna zawierać:

1. logo w kolorze i nazwę szkoły wg podanego wzoru (czcionka Times New Roman, bold, 16 pkt.),
2. nazwę wydziału, kierunek oraz specjalność (czcionka Times New Roman, bold, 16 pkt.),
3. imię i nazwisko autora (autorów) pracy, nr albumu (czcionka: Times New Roman, 15 pkt.) oraz tytuł pracy dyplomowej (czcionka: Times New Roman, *kursywa*, bold, 20 pkt.),
4. tytuł/stopień naukowy, imię i nazwisko promotora (czcionka: Times New Roman, 12 pkt.),

*UWAGA! Oprawa pracy dyplomowej zawiera okienko o wymiarach 15,5 x 6,5 cm, (umieszczone centralnie w stosunku do marginesów bocznych oraz w odległości 12,5 cm od dolnej krawędzi strony), w którym powinny się zmieścić informacje zawarte w pkt. c) i d).*

1. napis PRACA DYPLOMOWA MAGISTERSKA (lub LICENCJACKA/INŻYNIERSKA), (czcionka: Times New Roman, bold, 19 pkt),
2. napis Rzeszów oraz rok złożenia pracy dyplomowej, np. Rzeszów 2018 (czcionka: Times New Roman, bold, 14 pkt.).

Na kolejnych stronach znajduje się wzór strony tytułowej pracy dyplomowej - wersja dla prac przygotowanych w języku polskim (na przykładzie kierunku Informatyka) oraz wersja dla prac przygotowanych w języku angielskim (na przykładzie kierunku Filologia).

1. dziekan wydziału może w drodze decyzji określić bardziej szczegółowe wytyczne dotyczące zawartości pracy obowiązujące dla danego kierunku studiów [↑](#footnote-ref-1)