

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Івасів Володимир Антонович

**Система автоматизованого документообігу
перекладацького агентства / Automated document
circulation system of the translation agency**

спеціальність: 6.050102 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерні системи та мережі

Випускна кваліфікаційна робота

Виконав: студент групи КСМ-42/1
Івасів Володимир Антонович

Науковий керівник:
Ігнатєв І.В.

Випускну кваліфікаційну роботу
допущено до захисту:

" ___ " _____ 20__ р.

Завідувач кафедри
О. М. Березький

ТЕРНОПІЛЬ - 2019

РЕЗЮМЕ

Бакалаврська робота містить 69 сторінок пояснюючої записки, 21 рисуноків, 7 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуш формату А3.

Метою бакалаврської роботи є розробка системи автоматизованого документообігу перекладацького агентства.

Дана бакалаврської роботи присвячена огляду існуючих інструментів і технологій для розробки веб-сайтів. Розглядалися існуючі системи документообігу і наводився опис розробки веб-сайту. На основі цього було реалізовано розробку веб-сайту за допомогою технології Ruby On Rails і мови програмування Ruby. В роботі проведено аналіз основних вимог щодо структури сайту, розроблено загальний дизайн веб-сайту, алгоритм розподілу користувачів і адміністративну панель для роботи з замовленнями і документами.

Реалізовано систему автоматизовану документообігу перекладацького агентства для перекладу документів.

Ключові слова: СИСТЕМА АВТОМАТИЗОВАНОГО ДОКУМЕНТООБІГУ, АДМІНІСТРАТИВНА ПАНЕЛЬ, RUBY, RUBY ON RAILS.

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№. д о к у м.	П і д п	Д а		10

RESUME

Bachelor work contains 69 pages of explanatory note, 21 figures, 7 tables, 2 annexes. The volume of graphic material is 2 sheets of A3 format.

The purpose of the bachelor work is to develop a system of automated translation work of the translation agency.

This bachelor work is devoted to the review of existing tools and technologies for the development of websites. Existing systems of document circulation were considered and a description of the development of a website was given. Based on this, the development of the website was implemented using Ruby On Rails technology and Ruby programming languages. The paper analyzes the basic requirements for the structure of the site, develops the general design of the website, the algorithm of user distribution and the administrative panel for dealing with orders and documents.

The system of automated document circulation of the translation agency for translation of documents is implemented.

Keywords: AUTOMATED SYSTEM, ADMINISTRATIVE PANEL, RUBY, RUBY ON RAILS.

					Б Р . К С М. 07113/15.00.00.000. П 3	А в
З м	А в	№. д о к у м.	П і д п	Д а		11

ЗМІСТ

Вступ.....	8
1 Технології розробки і функціонування веб-технологій	10
1.1 Програмні засоби створення веб-сторінок	10
1.2 Вибір мови програмування для створення веб-сайту	12
1.3 Взаємодія веб-серверів з середовищем інтернет.....	16
1.4 Постановка задачі	18
2 програмні засоби розробки веб-сайту	19
2.1 Алгоритм роботи сайту.....	19
2.2 Алгоритм розподілу доступу.....	24
2.3 Структура сайту	26
3 Практична частина розробки веб-сайту	28
3.1 Розробка та створення WEB-дизайну	28
3.2 Back-end реалізація проекту	38
3.3 Тестування та верифікація.....	43
4 Техніко-економічне обґрунтування розробки комп'ютерної системи.....	47
4.1 Розрахунок витрат на розробку програмного забезпечення.....	47
4.2 Визначення експлуатаційних витрат.....	53
4.3 Розрахунок ціни споживання програмного продукту.....	55
4.4 Визначення показників економічної ефективності.....	57
4.5 Висновки.....	58
Висновки.....	59
Список використаних джерел.....	61
Додаток А Лістинг коду контролера OrdersController	63
Додаток Б Світокопія публікації	65
Додаток В Довідка про використання.....	Ошибка! Закладка не определена.

					БР.КСМ. 07113/15.00.00.000 ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Івасів В.А.			Літ.	Арк.	Акрушів
Перевір.		Ігнатев І.В.				8	69
Конс.					ТНЕУ. ФКІТ. КСМ-42/1		
Н. Контр.		Гураль І.В.			АГЕНТСТВА		
Затверд.		Березький О.М.					

ВСТУП

Перший веб-сайт був створений Тімом Бернерсом-Лі і розпочав свою роботу 6 серпня 1991 року. Він описував нові технології World Wide Web.

У наш час веб-сайти можуть виконувати дуже різні завдання, від пошуку потрібної інформації до покупок різних речей не виходячи з дому. Існують такі типи веб-сайтів: соціальні мережі, інтернет-магазини, файлообмінники, онлайн сервіси. Наявність сайту для великих організацій – це чудова можливість доносити велику кількість інформації до людей. За допомогою веб-сайту можна поширювати інформацію в будь-який час, швидко змінювати її, що дозволяє користувачам, які відвідують ваш сайт володіти найновішою інформацією. Жоден з мес-медіа не здатний на таке. Рекламу на радіо можна не почути або неправильно зрозуміти, а у газеті ми не можемо розмістити стільки важливої інформації. Присутність сайту в будь-якій організації не ставить перед нею ніяких обмежень і вона може розмістити велику кількість інформації. Сайт працює цілодобово і користувачі можуть відвідати його в зручний для них час. Найважливішим завданням веб-сайту є – донести інформацію до користувачів і потенційних клієнтів. Величезна кількість людей користується різними послугами і шукає потрібну їм інформацію в Інтернеті.

Дуже зручно, коли в вас є можливість перекладати ваші документи в електронному варіанті. Вам не потрібно ходити по кабінетах і чекати в чергах. Кожен клієнт користується такими послугами, щоб заощаджувати свій час, а це залежить від ефективності роботи веб-сайту відповідної організації. Робота з документами дуже важлива, як для звичайних клієнтів так і для самих організація. В організаціях робота з файлами повинна бути максимально ефективною, щоб кожен працівник не витрачав багато часу на роботу з документами.

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№. д о к у м .	П і д п	Д а		8

Підвищення ефективності управління документами в різних організаціях, де використовується система документообігу, а в даному випадку – сайту перекладацького агентства, багато в чому залежить від правильного ведення документів і взаємодії адміністратора з документами.

В кожній організації, де використовується система документообігу, документ є основним інструментом управління. Те саме стосується перекладацького агентства, якому важко працювати без наявної системи.

Перекладацьке агентство включає в себе основні операції з документами, наприклад:

- завантаження і видалення документів;
- розподіл документів між різними користувачами;
- індексація документів в системі;
- пошук документа за заданими параметрами.

Керування такою системою – трудомісткий процес і для легшої роботи з цією системою інтерфейс веб-сайту, адміністративної панелі, повинен бути інтуїтивно зрозумілий для адміністратора. Те саме стосується і клієнтів, в яких не повинно виникати ніяких проблем при замовленні послуг.

Метою бакалаврської роботи є аналіз інструментів для розробки веб-сайтів і розробка веб-сайту перекладацького агентства.

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		9

1 ТЕХНОЛОГІЇ РОЗРОБКИ І ФУНКЦІОНУВАННЯ ВЕБ-ТЕХНОЛОГІЙ

1.1 Програмні засоби створення веб-сторінок

У ХХІ столітті Інтернет – це величезна мережа, яка дозволяє не тільки знаходити і читати нову інформацію, але і спілкуватися з друзями, переглядати відео і фотографії, користуватися різними додатками і також розвивати свій бізнес за допомогою веб-сайту. Сьогодні в Інтернеті понад 1.5 мільярда веб-сайтів. Всі веб-сайти відрізняються один від одного своєю дизайном і функціоналом, але інструменти для створення їх веб-сторінок здебільшого однакові. Можна назвати три основних інструменти, які використовуються для створення веб-сторінок: HTML, CSS, JavaScript .

Сьогодні Основою будь-якого веб-сайту є веб-сторінка. Веб-сторінка – це текстовий документ, написаний мовою розмітки HyperText Markup Language(HTML). HTML використовується для відображення веб-сторінок в мережі Інтернет. Кожна сторінка вміщує ряд з'єднань, які називаються гіперпосиланнями. Ці гіперпосилання дають змогу нам перейти на іншу веб-сторінку того ж самого або іншого веб-сайту [9].

Усі сторінки, які ви бачите в інтернеті, написані за допомогою однієї чи іншої версії HTML. На даний момент версія HTML5 є основною, яка має багато нових можливостей, а також вона стабільно підтримується всіма сучасними веб-браузерами.

Як і будь-які інші мови розмітки чи мови програмування, HTML має свій особливий синтаксис, який дуже простий для вивчення. Основою цього синтаксису є HTML елементи, які складаються з тегів.

Загалом роль HTML полягає в інформуванні браузера відображати текст, зображення та інші форми мультимедіа на веб-сторінці.

Веб-сторінка може не тільки складатися з звичайного тексту, ми також можемо описати стиль веб-сторінки. В цьому нам допомагає Cascading Style Sheets(CSS). CSS допомагає зберігати інформаційний зміст документа окремо

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№. д о к у м.	П і д п	Д а		10

від деталей його відображення. Відображення деталей веб-сторінки відомі, як його стилі. Веб-розробник може зберігати стилі окремо від вмісту, щоб бути в змозі:

- уникати дублювання;
- зробити написання стилю простішим;
- використовувати той самий вміст з різними стилями для різних цілей.

Ваш веб-сайт може мати тисячі сторінок, які виглядають схожими. Використовуючи CSS, ви зберігаєте інформацію про стиль у загальних файлах, які використовують усі сторінки. Тобто, за допомогою CSS ми можемо створювати правила та застосовувати ці правила до багатьох елементів на веб-сайті. Коли користувач відкриває веб-сторінку, браузер користувача завантажує інформацію про стиль разом із вмістом сторінки. Оскільки стилі завантажуються лише один раз браузером, потім кешуються і використовуються для кожного завантаження сторінки, використання CSS може призвести до більш легких завантажень сторінок і покращення продуктивності. Це сприяє зменшенню навантаження на сервер і зниженню вимог, що в цілому економить гроші для власників веб-сайтів [4].

Оскільки ми можемо зберігати всі візуальні аспекти веб-сайту повністю відокремленими від вмісту, використовуючи CSS при розробці наших веб-сайтів, ми можемо швидко створювати макети та вирішувати проблеми. Незалежно від сторінки, ми контролюємо всі елементи з однієї (або декількох) таблиць стилів. Можна з упевненістю сказати, що CSS є невід'ємною технологією в розробці веб-сторінок і веб-сайтів.

На цьому веб-розробка не закінчується. Ще одним основним інструментом для створення веб-сторінок є JavaScript. JavaScript – це мова програмування на стороні клієнта, яка допомагає веб-розробнику займатися розробкою веб-додатків і створювати динамічні та інтерактивні веб-сторінки за допомогою користувацьких скриптів на стороні клієнта. JavaScript використовується для зміни вмісту веб-сайту та його поведінки різними способами у відповідь на дії користувача. Загальні використання

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№ д о к у м .	П і д п	Д а		11

JavaScript включають різні поля підтвердження, виклики до дії та додавання нових даних до існуючої інформації.Сьогодні JavaScript підтримується всіма сучасними веб-браузерами і використовується майже на кожному сайті в Інтернеті для більш потужної та складної функціональності [6].

Загалом JavaScript є більш складною мовою, ніж HTML або CSS. Більша частина динамічної поведінки, яку ви побачите на веб-сторінці, пов'язана з JavaScript, що збільшує контроль і поведінку браузера за замовчуванням.

1.2 Вибір мови програмування для створення веб-сайту

Створення веб-сайту дуже трудомісткий процес в якому беруть участь багато осіб. Якщо ви робите сайт для бізнесу, який буде мати складний функціонал і велику відвідуваність, то вам потрібна професійна команда. Тут потрібно розуміти, яка саме задача стоїть перед командою. Якщо задачею є створення простої веб-сторінки без зворотного зв'язку або сайт з малим функціоналом, то для цього не потрібна велика команда . Але якщо веб-проект є складним , в команді може працювати від 6 до 15 людей за такою спеціалізацією як:

- маркетолог;
- SEO-фахівець;
- UI / UX-дизайнер
- веб-дизайнер;
- автор тексту;
- Front-end розробник;
- Back-end розробник;
- контент менеджер;
- адміністратор проекту.

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№. д о к у м.	П і д п	Д а		12

Кожен з них спеціалізується на конкретному етапі і відіграє важливу роль в розробці веб-сайту, але можна відзначити двох спеціалістів без яких сайт неможливо створити. Це Front-end і Back-end розробники.

Користувачі бачать і взаємодіють тільки з тією частиною сайту, над якою працювали Front-end розробники. Все, що бачить користувач на сайті від шрифту, фону, меню, що випадає, слайдера створеного за допомогою HTML, CSS і JavaScript, контролюється браузером, який встановлений на комп'ютері користувача.

Front-end розробник створює “обличчя” сайту, частина, яку бачить користувач. Він враховує досвід взаємодії користувачів з сайтами. Щоб користувач був задоволений усім і розумів, як працювати з сайтом, розробнику необхідно знати мови програмування HTML, CSS і JavaScript, а також вміти працювати з фреймворком Bootstrap, AngularJS, бібліотеками типу jQuery і технологією Ajax, яка дозволяє звертатися до сервера без перезавантаження сторінки, а це живий пошук, елементи інтерфейсу, динамічне підвантаження сайтів [11].

Front-end розробник співпрацює з дизайнерами та UX-аналітиками над розробкою сайту. Йому важливо спілкуватися з іншими командами розробників, щоб краще розуміти цілі і потреби бізнесу і потім пропонувати рішення, які будуть краще відповідати цілям бізнесу.

А що ж змушує працювати сайт? Де збираються і знаходяться основні дані, реєстрації, замовлення користувачів на покупку і заявки на зворотний дзвінок? Back-end частина сайту складається з сервера, додатку і баз даних. Back-end розробник будує і підтримує технології, які керують усіма компонентами сайту. Він відповідає за весь внутрішній функціонал веб-сайту, бізнес логіку.

Для того, щоб сервер і бази даних працювали разом, Back-end розробник використовує серверні мови програмування на зразок PHP, Ruby, Python, Java, або .Net, інструменти MySQL, Oracle та SQL Server [11].

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		13

Перед тим, як приступити до роботи веб-розробники спілкуються з бізнес-аналітиками та стейкхолдерами проекту, щоб зрозуміти їх конкретні потреби і перевести це в технічні вимоги до сайту і підібрати найефективніші рішення і архітектуру.

Для створення веб-сайту, веб-розробнику не тільки потрібно знати мову програмування на якій потрібно писати логіку сайту, але й правильно використовувати цю мову з іншими інструментами і знати, як вони з нею працюють. На даний момент не існує будь-якої мови для веб-програмування, яка би переважала інші мови. Досвідчені програмісти віддають перевагу тій чи іншій мові, тільки виходячи з контексту поставленого завдання. Однак, це не є вимогою оскільки поставлене завдання можна вирішити будь-якою з популярних мов веб-програмування.

Дуже часто для створення веб-сайту, мову програмування використовують разом з фреймворком, що полегшує життя розробнику. Веб-фреймворк – інструмент, який полегшує процес написання і запуску веб-додатки. Вам не потрібно самостійно писати купу коду і витратити час на пошук потенційних прорахунків і помилок. Зараз їх різноманітність дозволяє вибрати відповідний як для статичних, так і для динамічних веб сторінок. Залежно від поставленого завдання, ви можете вибрати один фреймворк, що покриває всі потреби, або поєднати кілька.

У фреймворків є дві основні функції: робота на серверній стороні (Back-end) і робота на клієнтській стороні (Front-end).

Front-end фреймворки пов'язані з зовнішньою частиною програми. Простими словами, вони відповідають за зовнішній вигляд програми. Back-end відповідає за внутрішній функціонал додатку. Розглянемо обидва типи детальніше.

Клієнтські фреймворки. Цей тип фреймворків працює в браузері. З їх допомогою можна поліпшити і впровадити нові призначені для користувача інтерфейси. Front-end фреймворки дозволяють створювати різні анімації і

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		14

односторінкові додатки. Всі клієнтські фреймворки відрізняються по функціональності і використанню [10]. Ось деякі з них:

- Backbone + Marionette;
- Angular;
- Ember.js;
- Vue.js.

Всі вище перераховані фреймворки побудовані на основі JavaScript.

Серверні фреймворки. Правила та архітектура таких фреймворків не дає можливості створити веб-додаток з багатим інтерфейсом. Вони обмежені в своїй функціональності, проте ви все одно можете створювати прості сторінки і різні форми. Також вони можуть формувати вихідні дані і відповідати за безпеку в разі атак. Все це виразно може спростити процес розробки. Серверні фреймворки в основному відповідають за окремі, але критично важливі частини програми, без яких вони не змогли б нормально працювати. Ось кілька найпопулярніших фреймворків і мови, з якими вони працюють [10]:

- Django – Python;
- Laravel – PHP;
- Express.js – JavaScript;
- Ruby on Rails – Ruby.

Кожен з перелічених вище фреймворків є по-своєму особливий. Неможливо сказати, що якийсь з них є найкращим. Кожен веб-розробник вибирає для себе той фреймворк, який він найкраще знає або, який відповідає його задачам [5].

					Б Р . К С М . 07113/15.00.00.000 . П 3	А в
З м	А в	№ д о к у м .	П і д п	Д а		15

1.3 Взаємодія веб-серверів з середовищем інтернет

Веб-сервер – це система, яка надає контент або послуги кінцевим користувачам через Інтернет. Веб-сервер – основа Всесвітньої павутини. Веб-сервер складається з фізичного сервера, серверної операційної системи (ОС) і програмного забезпечення, що використовується для забезпечення зв'язку НТТР [2].

Основна мета веб-сервера полягає в тому, щоб зберігати файли веб-сайтів і транслювати їх по інтернету для відвідувачів сайту. По суті, веб-сервер – це просто потужний комп'ютер, який зберігає та передає дані через Інтернет. Коли хтось відвідує веб-сторінку на вашому сайті, їх браузер спілкується з вашим веб-сервером, надсилаючи та отримуючи інформацію, яка в кінцевому підсумку диктує те, що з'являється на екрані комп'ютера відвідувача. Таким чином, основною метою веб-сервера є зберігання та передача даних веб-сайту за запитом браузера відвідувача.

Кожному веб-серверу, який підключений до Інтернету, призначається ІР-адреса, яка ідентифікує цей пристрій в мережі. Навіть звичайні комп'ютери користувачів мають ІР-адресу, коли вони підключаються до Інтернету. Коли ви відвідуєте веб-сайт, від вашого комп'ютера з встановленою ІР-адресою надсилається НТТР-запит на ІР-адрес веб-сервера. Потім веб-сервер реагує запит, відсилаючи дані користувачу, який робив запит з відповідною ІР-адресою, і це процес, через який всі дані передаються на ваш комп'ютер з веб-серверу [1-2].

Більше 90% веб-сайтів в Інтернеті в даний час використовують орендований веб-сервер, який надається компанією з веб-хостингу.

Веб-хостинг – це послуга, яка дозволяє організаціям та приватним особам розміщувати веб-сайт або веб-сторінку в Інтернеті. Веб-хост або постачальник послуг веб-хостингу – це бізнес, який надає технології та послуги, необхідні для перегляду веб-сайту або веб-сторінки в Інтернеті. Веб-сайти розміщуються

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		16

або зберігаються на веб-серверах. Коли користувачі Інтернету хочуть переглядати ваш веб-сайт, все, що потрібно зробити, це ввести адресу веб-сайту або домен у свій веб-переглядач. Після цього їх комп'ютер з'єднається з сервером, а ваші веб-сторінки будуть доставлятися до них через веб-переглядач.

Доменне ім'я використовується для пошуку та ідентифікації комп'ютерів в Інтернеті. Доменне ім'я – це адреса, з якої користувачі Інтернету можуть отримати доступ до вашого веб-сайту. Комп'ютери використовують IP-адреси, які є набором чисел. Однак людям важко запам'ятовувати рядки чисел. У зв'язку з цим, доменні імена були розроблені та використані для ідентифікації об'єктів в Інтернеті, а не для використання IP-адрес. Перетворення IP-адреси в доменне ім'я здійснюється за допомогою DNS-сервера [2]. Більшість хостингових компаній вимагають, щоб ви мали власний домен для розміщення сайту. Якщо у вас немає домену, хостинг компанії допоможе вам придбати його.

На рисунку 1.1 зображено деякі функції, які веб-хостинг може надавати для клієнтів.



Рисунок 1.1 – Функції веб-хостингу

					Б Р . К С М . 07113/15.00.00.000. П 3	А в
З м	А в	№. д о к у м.	П і д п	Д а		17

Без веб-серверів інтернет, як ми знаємо, перестав би існувати. Веб-сервери є невід'ємною частиною роботи Інтернету. Індустрія веб-хостингу просто використовується для надання в оренду веб-серверів, що надає середнім власникам бізнесу та приватним особам можливість використовувати високотехнологічні сервери, які дозволяють розширювати свою діяльність на весь світ.

1.4 Постановка задачі

В сучасну епоху для всіх великих і малих підприємств дуже важливо мати свій власний веб-сайт, який відзначає їхню присутність в Інтернеті. Робота з документами присутня в усіх підприємствах, а швидка робота з ними покращує продуктивність працівників. В перекладацькому агентстві вся робота зосереджена над документами: правильним розміщенням документів в системі, найменування документів, редагування, конфіденційність інформації.

Метою бакалаврської роботи є створення системи документообігу перекладацького агентства. Одним з основних завдань, є створення веб-сайту для перекладу документів. Перевагою цього сайту повинно бути швидкий і якісний переклад документів серед існуючих аналогів. Кожен користувач буде мати змогу в простій для нього формі зробити замовлення на переклад документа.

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		18

2 ПРОГРАМНІ ЗАСОБИ РОЗРОБКИ ВЕБ-САЙТУ

2.1 Алгоритм роботи сайту

Перш ніж почати створювати власний веб-сайт і запускати його в інтернеті, важливо знати, як працює веб-сайт.

Ось деякі основні терміни, які потрібно знати щоб пояснити роботу веб-сайту:

- веб-сайт – це набір веб-сторінок, що описують макет, формат і вміст сторінки;
- веб-сервер – це комп'ютер підключений до Інтернету на якому розміщений ваш веб-сайт, який отримує запит на веб-сторінку, надіслану вашим веб-переглядачем;
- браузер(веб-переглядач) з'єднується з сервером через IP-адресу. IP-адреса отримується шляхом перетворення доменного імені [12].

Веб-сайт, як правило, являє собою набір веб-сторінок, зображень та інших елементів, які пов'язані між собою для формування великого, структурованого документа. Думайте про веб-сайт як про книгу, а кожна окрема сторінка – це веб-сторінка.

Веб-сайт може складатися з однієї сторінки або може містити тисячі сторінок. Кожна веб-сторінка має власний текст, зображення та інші елементи.

Кожна веб-сторінка написана мовою розмітки, і ця мова описує макет, формат і вміст сторінки. Найпоширенішою мовою розмітки для створення веб-сторінок є HTML. Всі сторінки веб-сайту, як і інша інформація зберігається на веб-сервері.

Веб-сервер – це комп'ютер, який приймає HTTP запити, що відправляється вашим браузером. Основна мета веб-сервера полягає в тому, щоб зберігати, обробляти і доставляти веб-сторінки користувачам. Цей взаємозв'язок виконується за допомогою протоколу передачі гіпертексту (HTTP). Ці веб-сторінки в основному є статичним контентом, який включає

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№ д о к у м .	П і д п	Д а		19

HTML-документи, зображення, таблиці стилів, тексти і т.д. Крім HTTP, веб-сервер також підтримує протокол SMTP (Simple Mail transfer Protocol) і FTP (File Transfer Protocol) для надсилання електронної пошти і для передачі і зберігання файлів.

Кожна веб-сторінка в Інтернеті також має унікальну адресу, яка називається Uniform Resource Locator або URL. Наприклад, www.example.com/page1.htm є типовим URL-адресою. Коли користувач вводить цей URL у веб-переглядач, комп'ютер, на якому працює браузер, надсилає запит на IP-адресу комп'ютера, на якій запущено веб-сервер для цієї сторінки, з проханням надіслати весь знайдений вміст назад. Після того, як веб-сервер отримає цей запит, він відправляє вміст сторінки назад по IP-адресі комп'ютера, який відправляв запит.

Для того щоб наші URL запити і назва веб-сайту були зрозуміліші, використовують доменне ім'я.

Доменні імена вигадали, щоб було зручніше знаходити сайти в інтернеті. Раніше потрібно було запам'ятати цифрову адресу сайту, щоб відкрити його у веб-переглядачі. Цифрова чи IP-адреса – це комбінація з чотирьох чисел, розділених крапками. IP-адреса вказує на сервер, де розташований сайт, і має такий вигляд: 7.135.10.13 [13].

Можна запам'ятати кілька таких номерів, але спробуйте втримати в голові кілька десятків IP-адрес, і числа почнуть плутатися. Комп'ютерам усе одно, вони запам'ятають скільки завгодно чисел, а для людей вигадали текстові імена. Тому спеціальні DNS сервери перетворюють вашу IP адресу на звичайну назву, зрозумілу кожному користувачу Інтернет [13].

Будь-яка інформації, яка передається між браузером і веб-сервером в мережі здійснюється на основі протоколу HTTP. HTTP (Hypertext Transfer Protocol) – це набір правил для передачі файлів (текстових файлів, графічних зображень, звуку, відео та інших мультимедійних файлів) у World Wide Web . Як тільки користувач Інтернет відкриває свій веб-браузер , він неявно використовує HTTP. Коли користувач переходить по будь-якій адресі, він здійснює HTTP

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		20

запит. У цій ситуації ваш комп'ютер, який робить запит, називається клієнтом. URL-адреса, яку ви запитуєте, це адреса, яка належить серверу [14].

Після встановлення з'єднання TCP , клієнт надсилає на сервер запит HTTP GET, щоб отримати веб-сторінку, яка повинна йому відобразитися. Після того, як сервер відправив відповідь, він закриває з'єднання TCP. Якщо ви знову відкриєте веб-сайт у вашому браузері, або якщо ваш браузер автоматично запитує щось в сервера, відкривається нове з'єднання, яке слідує описаному вище процесу. Запити GET є одним з видів методу HTTP, який клієнт може викликати. Є також інші методи: POST, PUT, DELETE. Ці всі методи представлені в одній архітектурі, яка називається REST API. На рисунку 2.1 представлено приклад використання HTTP запитів з відповідними URL-адресами за допомогою фреймворка Ruby on Rails.

Routes

Routes match in priority from top to bottom

Helper	HTTP Verb	Path
Path / Url		<input type="text" value="Path Match"/>
new_user_session_path	GET	/users/sign_in(:format)
user_session_path	POST	/users/sign_in(:format)
destroy_user_session_path	DELETE	/users/sign_out(:format)
new_user_password_path	GET	/users/password/new(:format)
edit_user_password_path	GET	/users/password/edit(:format)
user_password_path	PATCH	/users/password(:format)
	PUT	/users/password(:format)
	POST	/users/password(:format)

Рисунок 2.1 – HTTP запити в Ruby on Rails фреймворку

Ми можемо побачити , що кожен URL має свій HTTP запит і відповідно ця URL вказує шлях до сторінок веб-сайту.

Є також інший протокол, HTTPS. HTTPS (Hypertext Transfer Protocol Secure) – це безпечна версія HTTP, яка є основним протоколом, який використовується для передачі даних між веб-браузером і веб-сайтом. HTTPS шифрується для підвищення безпеки передачі даних. Це особливо важливо, коли користувачі передають конфіденційні дані, наприклад, вхід до банківського рахунку, служби електронної пошти або постачальника медичного страхування. Будь-який веб-сайт, особливо ті, які потребують облікових даних для входу, повинні використовувати HTTPS [15].

HTTPS використовує протокол шифрування для шифрування повідомлень. Протокол називається Security Layer Transport (TLS), хоча раніше він був відомий як Secure Sockets Layer (SSL). Цей протокол забезпечує зв'язок за допомогою так званої асиметричної інфраструктури відкритого ключа. Цей тип системи безпеки використовує два різних ключа для шифрування зв'язку між двома сторонами: закритий та відкритий.

Закритий ключ – цей ключ контролюється власником веб-сайту. Цей ключ живе на веб-сервері і використовується для розшифрування інформації, зашифрованої відкритим ключем.

Відкритий ключ – цей ключ доступний кожному, хто хоче взаємодіяти з сервером безпечним способом. Інформація, зашифрована відкритим ключем, може бути розшифрована лише за допомогою закритого ключа.

З технічної точки зору, HTTPS не є окремим протоколом від HTTP. Це просто використання шифрування TLS / SSL на протоколі HTTP. HTTPS відбувається на основі передачі сертифікатів TLS / SSL, які перевіряють, що певний провайдер є тим, ким він є насправді.

Переглянувши основні аспекти роботи веб-сайту можна навести загальний алгоритм його роботи на основі шаблону MVC Ruby on Rails фреймворка (рисунок 2.2).

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		22

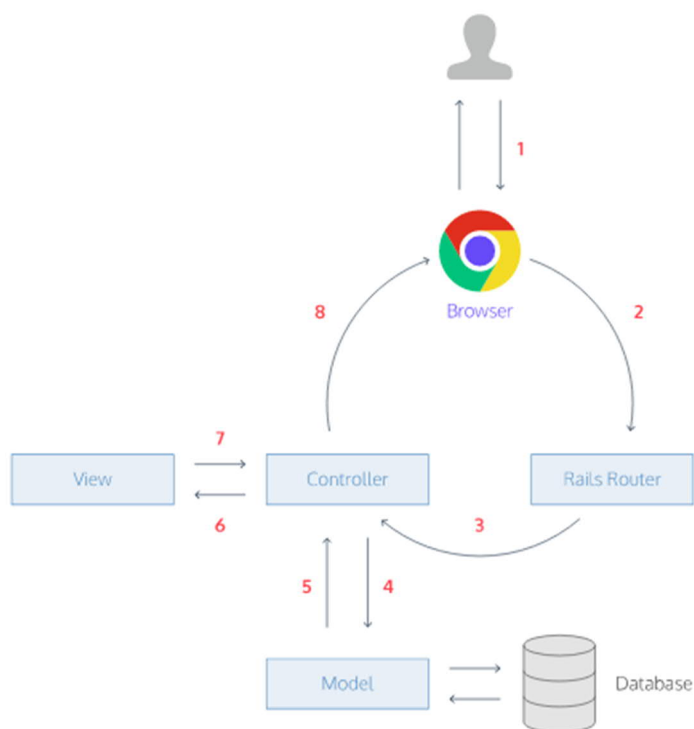


Рисунок 2.2 – Алгоритм запиту/відповіді на основі MVC

Даний алгоритм MVC, який представлений вище, працює наступним чином:

- користувач відкриває свій браузер, вводить URL і натискає Enter. Коли користувач натискає Enter, браузер робить HTTP запит до веб-сервера на цю URL-адресу;
- запит потрапляє на маршрутизатор Rails (config / routes.rb);
- маршрутизатор відображає URL-адресу правильного контролера і екшена для обробки запиту;
- екшен отримує запит і просить модель отримати дані з бази даних;
- модель повертає список даних до дії контролера;
- дія контролера передає дані на перегляд;
- перегляд відображає сторінку як HTML;
- контролер посилає HTML назад у браузер. Сторінка завантажується, і користувач бачить її.

В залежності від типів запиту до бази даних використовують різні HTTP запити.

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№. д о к у м.	П і д п	Д а		23

2.2 Алгоритм розподілу доступу

Сайт – це, як журнал і магазин. Це дозволяє відвідувачам бачити вас у віртуальному просторі, щоб вони могли робити конкретні припущення та створювати певне сприйняття вашого бренду. Веб-сайт – це чудовий інструмент, який потребує кожен бізнес. Однак більшість підприємців, які починають свій перший бізнес, можуть відкласти своє рішення про створення сайту.

У більшості випадків великі компанії мають веб-сайти, але не бачать потреби в них, тому що вони, здається, не отримують результатів від своїх веб-сайтів. Тому дуже важливо розуміти для чого вам потрібен сайт, які користувачі будуть ним користуватися і як вони повинні взаємодіяти з веб-сайтом. Власник сайту повинен пояснити веб-розробнику хто може користуватися їхніми послугами, скільки людей потрібно для адміністрування сайту і загальний бюджет. Для визначення чітких задач і цілей бізнес-проекту, веб-розробник з директор компанії створюють технічне завдання. В технічному завданні формулюються терміни і етап розробки.

Відповідно до поставленої задачі, веб-сайт перекладацького агентства повинен мати такі рівні доступу: звичайний користувач Інтернет, адміністратор і суперкористувач. Відповідно, кожна роль має обмежений доступ прав. Давайте розглянемо основні задачі суперкористувача.

Суперкористувач – найважливіша особа, яка має повний контроль над веб-сайтом. Зазвичай цією особою є програміст. Програміст знає про всі деталі і особливості вашого веб-сайту, може редагувати контент і змінювати внутрішній функціонал, логіку. На сайті Translation, програміст виконує задачі ті ж самі функції. Він також буде завантажувати сайт на веб-сервер і матиме загальний доступ до хостингу. Часто ці функції виконує адміністратор, але в даному випадку він відіграє трохи іншу роль.

Роль адміністратора в перекладацькому агентстві полягає в тому, щоб

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№ д о к у м .	П і д п	Д а		24

спілкуватися з клієнтами та перекладачами. Адміністратор керує всіма замовленнями користувача, а саме документами. Всі документи доступні йому у вигляді списку, і може виконувати будь-які функції над документом крім редагування. Важливо зазначити, що адміністратор повинен бути перекладачем і повинен мати професійні навички роботи з документами для перекладу. Адміністратор повинен правильно визначити складність перекладу і його вартість, після чого він надає ці дані перекладачу. Перекладач – це особа, яка здійснює переклад. Він має можливість спілкуватися з адміністратором, але він немає ніяких прав доступу на сайті, тільки якщо адміністратор йому їх не надасть.

Адміністратор сайту перекладацького агентства є важливим працівником, на якому лежить велика відповідальність і який виконує багато роботи. Крім цього, адміністратор буде керувати і розвивати блог сайту. Він може публікувати статті перекладачів, можливо і звичайних користувачів. Як керівник блогу, адміністратор має усі права для створення, видалення, редагування статей.

Простота в користуванні сайтом і швидкість замовлення перекладу є важливою для користувачів. Будь-який користувач в правах переглядати інформацію про підприємство на веб-сайті. Оскільки основною задачею сайту є переклад документів, то будь-якому клієнтові замовлення перекладу не повинно складати ніяких проблем. Відправивши коротку форму клієнт може дізнатися скільки часу займе його переклад і саму ціну, яка визначатиметься адміністратором. Також користувачі, які пройшли авторизацію мають можливість поділитися своїми думками в коментарях.

Дуже важливим є визначення прав доступу різних користувачів до окремих частин веб-сайту. Ось декілька причин чому важливо зробити розподіл доступу:

- веб-розробник матиме змогу швидко розробити веб-сайт;
- правильне розподілення прав доступу дає чітку структуру веб-сайту;
- це може спростити алгоритм роботи сайту.

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№. д о к у м.	П і д п	Д а		25

2.3 Структура сайту

Зовнішній вигляд кожного веб-сайту є унікальним, проте в усіх веб-сайтах можна знайти спільні частини. Коли користувачі заходять на будь-який сайт, вони потрапляють на головну сторінку. Її розробці приділяють особливу увагу, тому що тут відображається основна інформація для користувачів. Якщо потрібної інформації не має або вона не правильно структурована користувач залишає сайт. Щоб таких ситуацій не було потрібно правильно структурувати ваш веб-сайт.

Структура сайту - це схема розміщення основних розділів і сторінок одна відносно одного. Це план-схема, яка показує, яким чином будується сайт, логічна зв'язка його сторінок. Хороша структура сайту значно підвищує шанси на високий рейтинг у пошукових системах. Структура сайту залежить від того, який вид сайту очікує замовник[3].

В основному, у верхній частині головної сторінки розташоване навігаційне меню, що дублюється на інших сторінка веб-сайту. Таке розташування відразу допомагає користувачу на основі свого вибору легко переміщатися між сторінка. Такий розподіл на категорії, як правило називається внутрішню структурою. Приклад внутрішньої структура сайту зображено на рисунку 2.3.

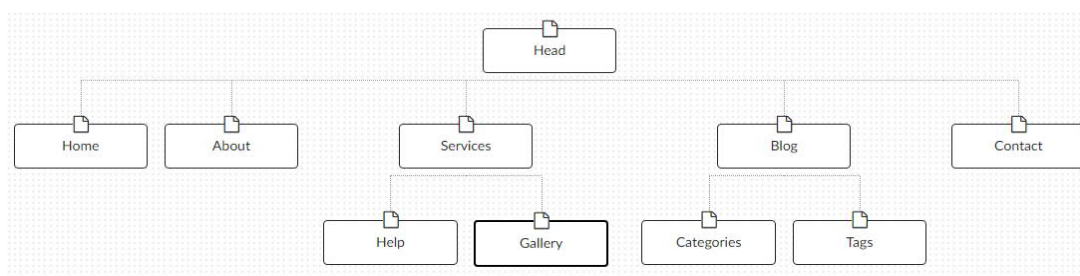


Рисунок 2.3 – Структура веб-сайту

Зовнішня структура сайт – це макет сторінок, різне розташування

інформаційних блоків або форм. Приклад зовнішньої структура сайту зображено на рисунку 2.4.

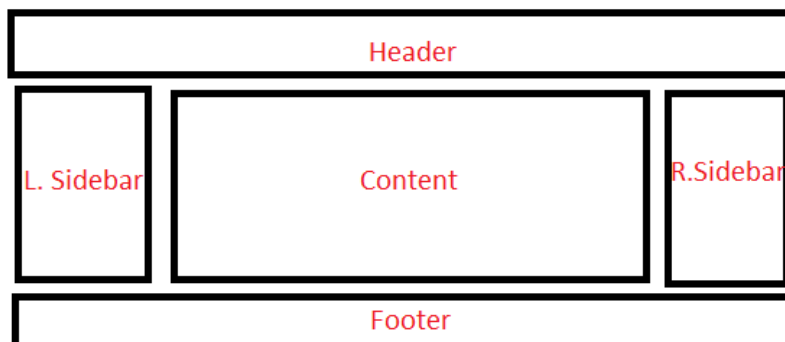


Рисунок 2.4 – Структура веб-сайту

Як ми вже говорили основною сторінкою сайту є головна сторінка, на якій користувач може переміщатися між різними розділами і підрозділами. Найчастіше на цій сторінці відображається інформація про підприємство, його переваги і можливості.

Іншими не менш важливими є сторінки “Про нас” і “Контакти”, де відображається коротка інформація про компанію, в якому році розпочалася її роботи, інформація про контактні дані, такі як телефон, емейл.

Загалом , структура веб-сайту повинна бути дуже простою. Не потрібно наповнювати сайт великою кількістю інформації, тому що для користувачів веб-сайт - це як реклама і вони зазвичай лише поверхово проглядають її.

3 ПРАКТИЧНА ЧАСТИНА РОЗРОБКИ ВЕБ-САЙТУ

3.1 Розробка та створення WEB-дизайну

Все, що стосується веб-сайту, у тому числі його вміст, вигляд, спосіб його роботи – визначається дизайном веб-сайту. Веб-дизайн – це поєднання зручності в користуванні і стильного візуального рішення, асоціативно відповідного іміджу і діяльності вашої компанії.

Кожен створений веб-сайт повинен мати унікальне дизайнерське рішення, адже саме дизайн робить його унікальним і несхожим на інші сайти. Дизайн сайту – це обличчя вашої фірми в Інтернеті. За якістю дизайну сайту визначають солідність і динамічність компанії, тому навіть невелика фірма маючи якісний сайт, може викликати більшу довіру у користувачів Інтернету, ніж велика компанія з неякісним сайтом. Все повинно бути виконано таким чином, щоб навіть випадковий відвідувач, зайшовши на вашу сторінку був би в захваті від вашого сайту.

Багато хто сприймає веб-дизайн та розробку сайту лише, як гарне оформлення сторінок. Але таке міркування не враховує такі важливі умови якісного дизайну, як функціональність навігації і структура розробленого сайту, висока якість графіки, зручне управління створеним сайтом і можливості його розширення, оптимізацію для пошукових систем. І лише врахувавши ці та багато інших чинників веб-дизайнер може гарантувати, що розробка і створення веб-сайту повністю задовольняє умови замовника, як по функціоналу, так і по дизайну. Головне завдання дизайну будь-якого сайту - викликати довіру відвідувача до фірми, до якості пропонованих нею товарів і послуг, зберігаючи при цьому фірмовий стиль, корпоративну культуру, і позиціонування компанії на ринку.

Веб-дизайн сайту і його побудова поєднує в собі різні нюанси, які необхідно враховувати при розробці та створенні веб-сайту. Перш за все це наявність зручної навігації та прозорості структури сайту.

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		28

3.1.1 Структура і макет веб-сайту

Для створення веб-сайту потрібно знати його структуру і створити макет. На основі цього, розробка веб-сайту буде простішою і допоможе дотримуватися поставленого завдання. Проаналізувавши наявність усіх інструментів для створення макету, Figma є найдоступнішим. Figma дозволяє розробляти прототип веб-сайту дуже швидко в режимі реального часу і в одному місці.

Самою основною сторінкою кожного веб-сайту є головна сторінка або як її ще називають домашня. Для веб-користувача домашня сторінка є першою веб-сторінкою, яка відображається після завантаження веб-сайта.

Немає стандартного макета домашньої сторінки, але більшість домашніх сторінок містить панель навігації(header), яка містить посилання на різні розділи веб-сайту. Інші загальні елементи, знайдені на домашній сторінці, включають інформацію про веб-сайт і посилання на іншу інформацію. На сайті Translations на головній сторінці розміщується основні блоки інформації. В цих блоках є тільки основні дані, щоб отримати більше інформації, користувач може перейти за з посиланням “Читати більше”.

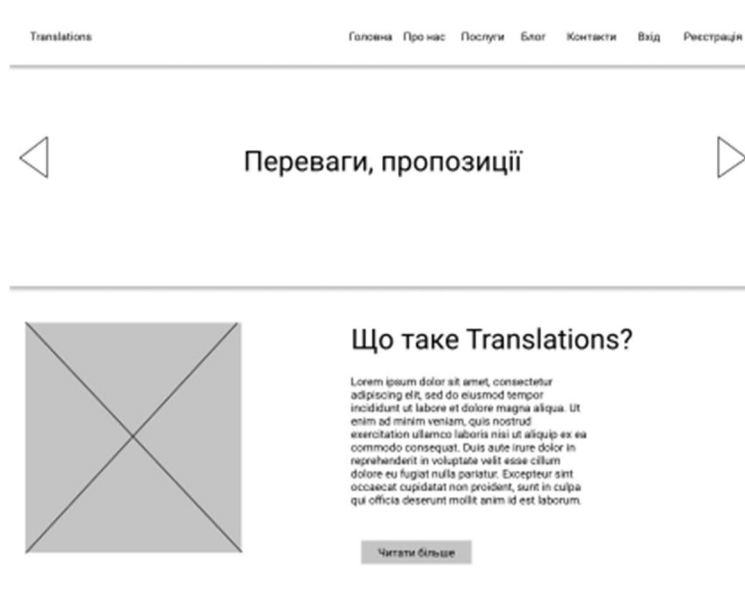


Рисунок 3.1 – Верхня частина макету домашньої сторінки

На рисунку 3.1 зображено макет головної сторінки веб-сайту Translations. В верхній частині сайту розміщується навігаційне меню, яке відіграє дуже важливу роль для користувачів веб-сайт. Це меню дозволяє користувачам переходити між різними розділами веб-сайту і постійно бачити його, оскільки воно закріплене, що означає постійну присутність навіть коли користувач переходить на інший розділ або прокручує веб-сайт донизу. Перший блок зображено у вигляді слайдера і дозволить користувачу бачити різну інформацію на одному блоці.



Рисунок 3.2 – Макет домашньої сторінки

Друга частина домашньої сторінки, яка зображена на рисунку 3.2 відобразить інформацію про послуги у вигляді чотирьох малих блоків з текстом.

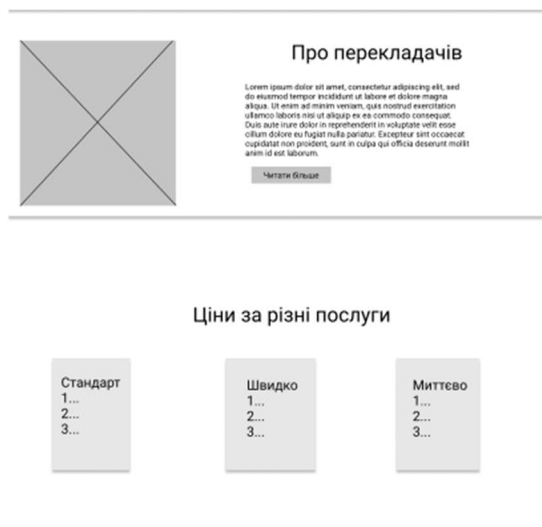


Рисунок 3.3 – Макет домашньої сторінки

З м	А р	№. д о к у м.	П і д п	Д а

На рисунку 3.3 зображено продовження макету домашньої сторінки. В блоці про перекладачів є посилання на детальнішу інформацію, яке перенаправляє користувача на сторінку “Про нас”.

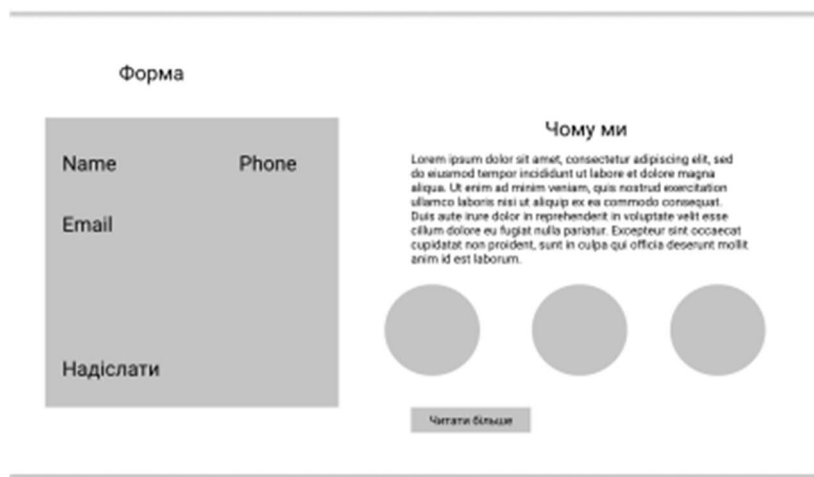


Рисунок 3.4 – Макет домашньої сторінки

Таку ж саму функцію виконує посилання в останньому блоці на рисунку 3.4, де також є форма для замовлення перекладу документів. Як ми можемо побачити з рисунків інформація в деяких блоках однакова. На даному етапі не важливо якою інформацією наповнений веб-сайт. Зараз найважливіше побудувати структуру, на основі якого можна приступати до розробки веб-дизайну.

На рисунку 3.5, кінцевому макеті домашньої сторінки, зображено блок з інформацією про те, як працює прийняття замовлень на даному веб-сайті. В останньому блоці “Footer” є додаткова інформація про веб-сайт і посилання на різні соціальні мережі. В цьому блоці може бути поле для отримання підписки, за допомогою якої ви можете отримувати усі свіжі новини, які стосуються компанії .

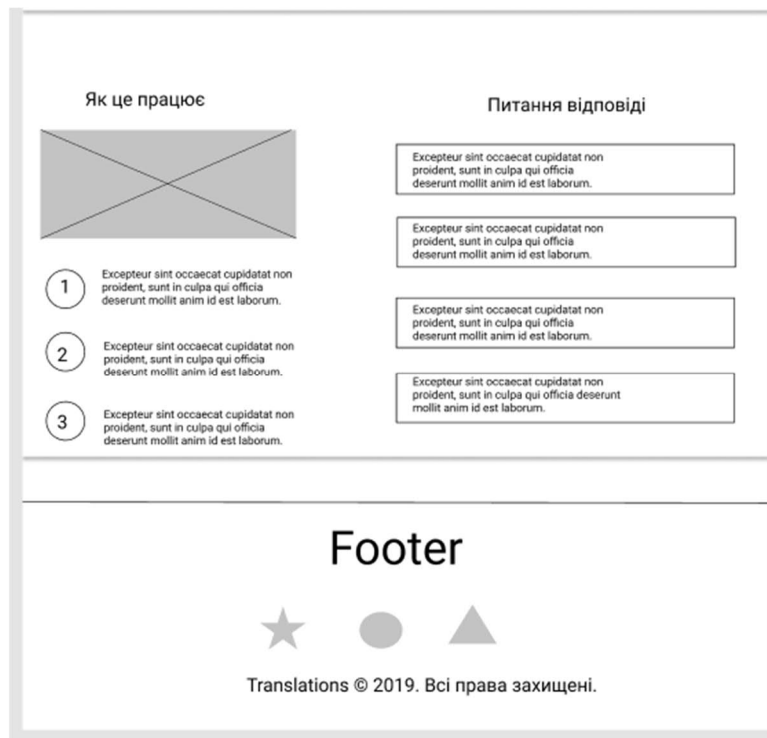


Рисунок 3.5 – Макет домашньої сторінки

Сторінка “Про нас”, “Послуги” мають схожу структуру блоків, але з різною інформацією. Сторінка “Контакти” і “Блог” відрізняється від попередніх.

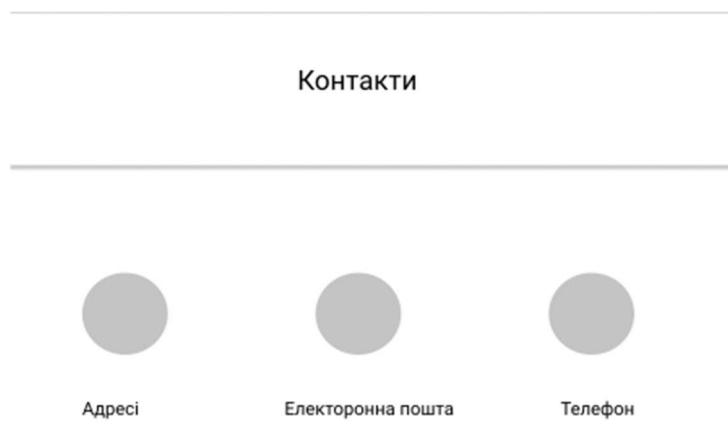


Рисунок 3.6 – Верхня частина макету сторінки “Контакти”

На рисунку 3.6 зображено макет блоку сторінки на якому відображається контактні дані компанії. Це може бути адреса електронної пошти, адреса розміщення офісу, а також номер телефону.

Карта з розміщенням офісу

Форма для відправлення повідомлень

Рисунок 3.7 – Нижня частина макету сторінки “Контакти”

На рисунку 3.7 зображено нижню частину сторінки “Контакти”. В цьому блоці відобразатиметься форма для відправлення повідомлень.

Остання сторінка, яка є не менш важливою – це сторінка “Блог”. Блог є тією частиною сайту, на якій розміщують цікаві статті на різні теми. Керування блогом здійснюється адміністратором сайту. Він може створювати, редагувати, оновляти і видаляти статті. Блог є важливою частиною зі сторони SEO-оптимізації. Це дозволяє пошуковим системам загрузити веб-сайт першими. Макет сторінки “Блог” зображено на рисунку 3.8.

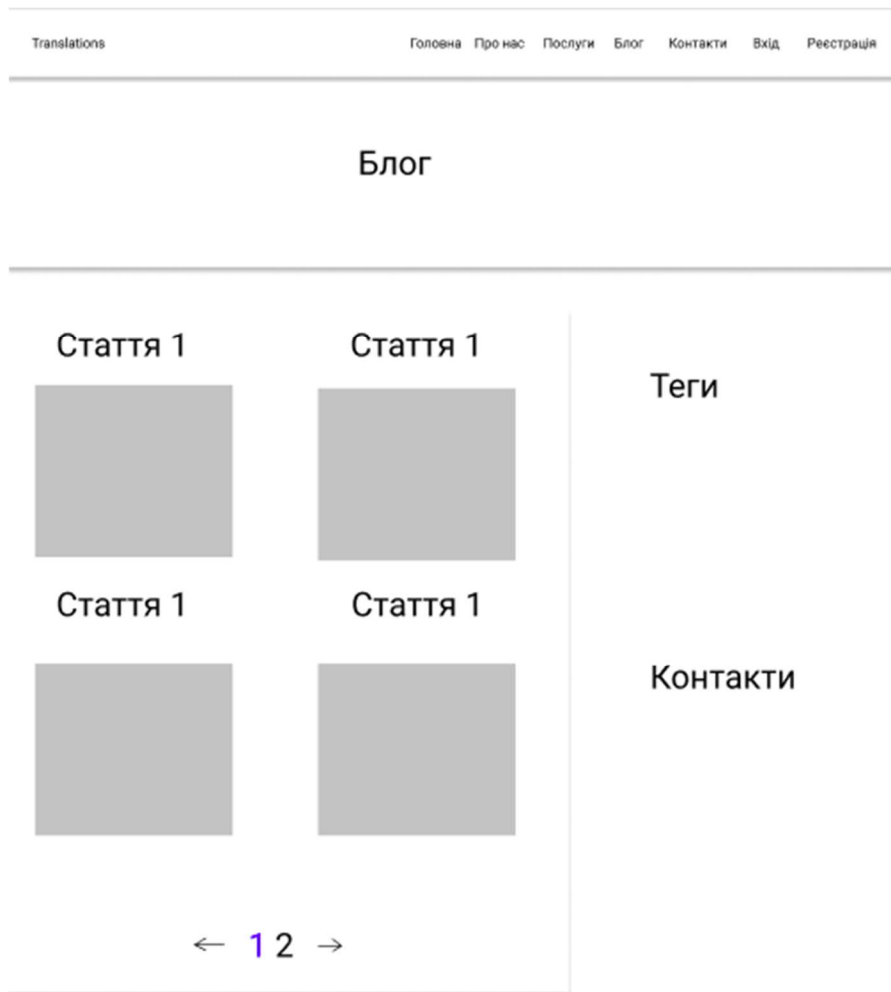


Рисунок 3.8 – Макет сторінки “Блог”

Веб-сайт Translations також дозволяє користувачам реєструватися і має ще дві сторінки для реєстрації і входу. Авторизація використовується для того, щоб користувач мав змогу залишати коментарі до будь-якої статті.

Після створення основного макету сайту, можна приступати до розробки його веб-дизайну.

3.1.2 Створення веб-дизайну за допомогою Bootstrap

Bootstrap – це найбільш популярний HTML, CSS, JS фреймворк для розробки адаптивних веб-сайтів. Bootstrap дозволяє веб-розробнику не тільки швидко і просто розробляти веб-додаток, а й не задумуватися над розробкою додаткових функцій для адаптивності під різні пристрої. Це пояснюється тим, що веб-сайти, створені за допомогою Bootstrap, автоматично масштабуватимуться між пристроями, незалежно від того, чи це мобільний телефон, планшет, ноутбук, настільний комп'ютер.

Bootstrap включає такі компоненти, як кнопки, навігаційні панелі, випадаючі меню, вікна сповіщення та багато іншого. У більшості випадків ви можете використовувати компонент просто за допомогою відповідного імені класу.

Оскільки веб-сайт Translations розробляється на фреймворку Ruby on Rails, в такому випадку існує окреме середовище розробки і для того щоб підключити Bootstrap до нашого додатку потрібно виконати наступні дії:

- встановлення gem Bootstrap і jQuery до нашого додатку;
- встановлення залежностей в файл application.js, які потрібні для коректної роботи Bootstrap;
- імпортування Bootstrap в файл стилів.

Після цього можна приступати до розробки веб-дизайну. Для прикладу на рисунку 3.8 зображено сторінку для реєстрації користувачів.

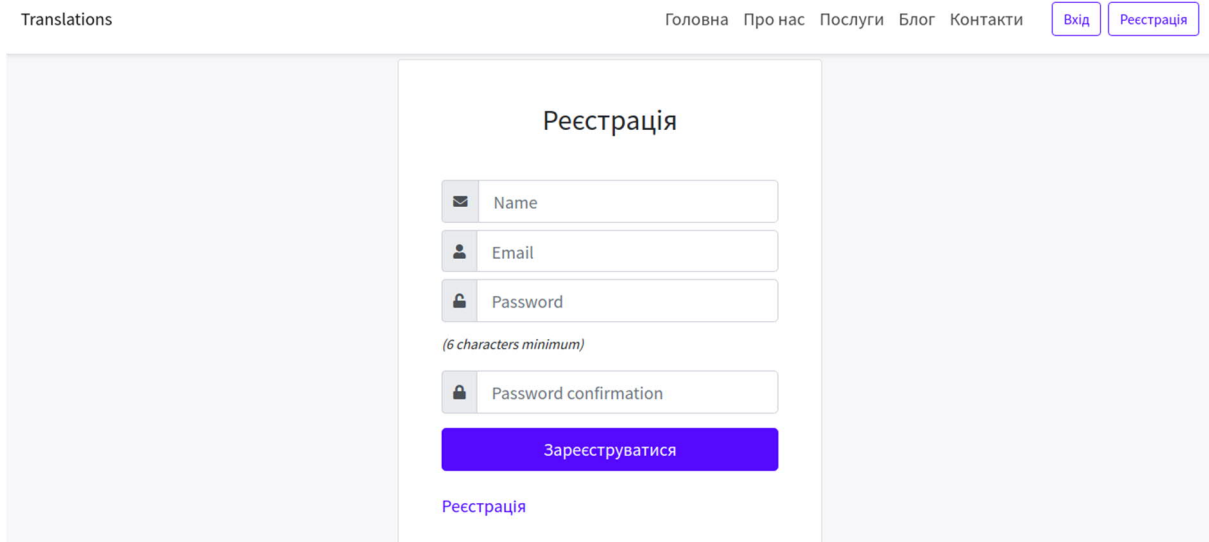


Рисунок 3.9 – Сторінка реєстрації

Нижче наведено код зовнішнього вигляду головної сторінки за допомогою Bootstrap:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
    <title>Translations</title>
    <%= csrf_meta_tags %>
    <%= csp_meta_tag %>
    <%= stylesheet_link_tag 'application', media: 'all', 'data-
turbolinks-track': 'reload' %>
    <link
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,
400,600" rel="stylesheet">
    <%= javascript_include_tag 'application', 'data-turbolinks-
track': 'reload' %>
  </head>
  <body>
    <%= render "shared/navbar" %>
    <div class="mt-78"></div>
    <%= render "shared/notice" %>
    <%= yield %>
    <%= render "shared/aside" %>
    <%= render "shared/footer" %>
  </body>
</html>
```

Код форми входу користувачів наведено нижче:

```
<%= form_for(resource, as: resource_name, url:
session_path(resource_name)) do |f| %>

  <div class="input-group mb-2 mr-sm-2">
    <div class="input-group-prepend">
      <div class="input-group-text"><%= icon('fas',
'user') %></div>
    </div>
    <%= f.email_field :email, autocomplete: "email",
placeholder: "Email", class: 'form-control form-control-lg' %>
  </div>

  <div class="input-group mb-2 mr-sm-2">
    <div class="input-group-prepend">
      <div class="input-group-text"><%= icon('fas',
'lock') %></div>
    </div>
    <%= f.password_field :password, autocomplete:
"current-password", placeholder: "Password", class: 'form-control
form-control-lg' %>
  </div>
  <% if devise_mapping.rememberable? %>
    <div class="form-group">
      <%= f.check_box :remember_me %>
      <%= f.label :remember_me %>
    </div>
  <% end %>
  <%= f.submit "Log in", class: 'btn btn-lg btn-primary
btn-block mb-4' %>

<% end %>
```

В коді зовнішнього вигляду можна побачити, що на основі елементів `div` і різних класів будується Bootstrap сітка. Ця сітка ділиться на дванадцять рівних частин за допомогою класу `col` і відповідно до кількості блоків розраховується пропорція, що розділяє веб-сайт або його структуру на Grid сітку. Це дозволяє швидко розробляти веб-сайт. Також не потрібно стилізувати розмітку сайту, Bootstrap допомагає зробити це замість веб-розробника.

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		37

3.2 Back-end реалізація проекту

Для реалізації веб-сайту Translations використовується фреймворк Ruby On Rails. Ruby on Rails (RoR) – це багаторівневий фреймворк для побудови веб-додатків, який використовує реляційні та NoSQL бази даних (наприклад, MySQL, MariaDB, PostgreSQL, MongoDB). Фреймворк написаний на мові програмування Ruby. Rails підходить як для розробки звичайних сайтів, які повинні бути дійсно швидкими, відмовостійкими і працювати під високим навантаженням, так і для веб-додатків з складною бізнес-локальною і динамічним веб-інтерфейсом. Ruby on Rails є відкритим програмним забезпеченням і розповсюджується під ліцензією MIT [16].

На рисунку рисунок 3.10 зображено загальну картину роботи веб-програми на основі фреймворку Ruby on Rails.

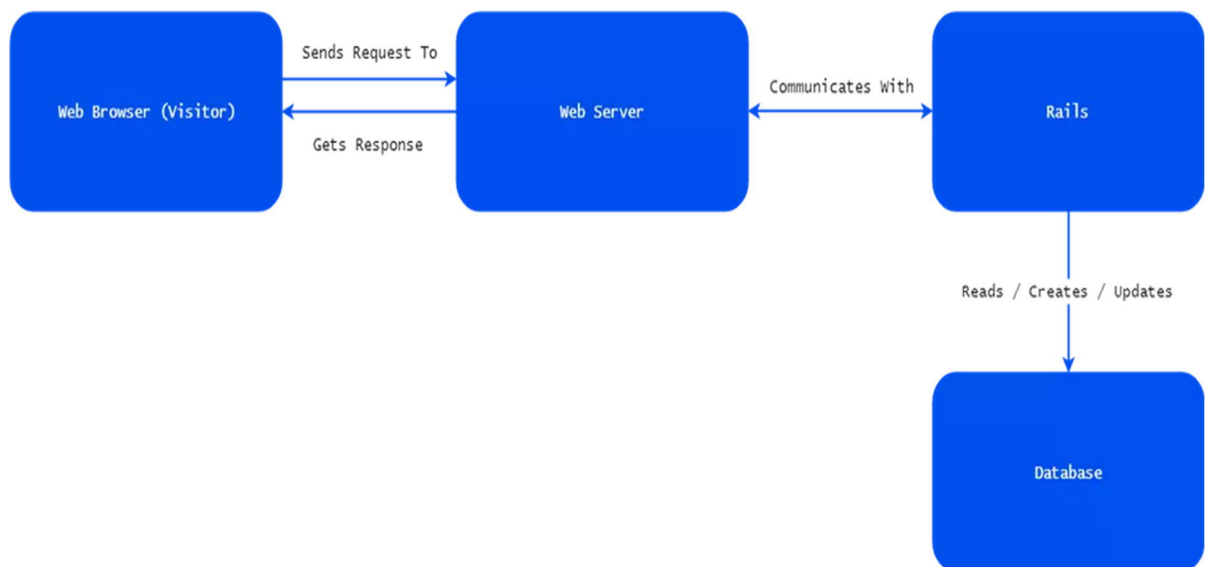


Рисунок 3.10 – Принцип роботи веб-програми на RoR

Rails отримує запити, направляє їх у відповідні екшени, які потім взаємодіють з базою даних (через ActiveRecord) для виконання запиту. Потім база даних повертає результати (HTML або JSON) користувачеві.

Rails використовує архітектуру MVC .Model-View-Controller (MVC) є архітектурним шаблон , який розділяє додаток на три основні логічні компоненти: модель, вигляд і контролер. Кожен з цих компонентів побудований для обробки конкретних аспектів розробки програми. MVC є одним з найбільш часто використовуваних галузевих стандартів веб-розробки для створення масштабованих і проектів з можливістю розширення. Структура MVC шаблону зображено на рисунку 3.11.

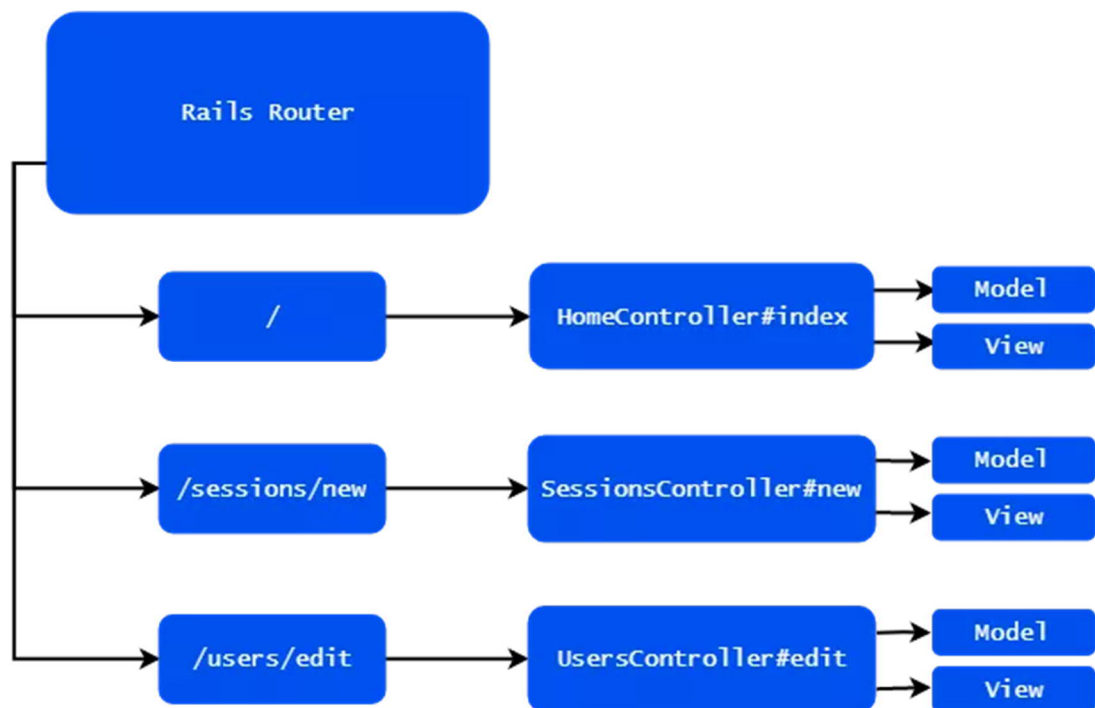


Рисунок 3.11 – Структура архітектури MVC

Екшени організовані в контролерах, контролери приймають рішення про те, як обробляти запит, і вони звертаються до бази даних за будь-якою потрібною інформацією. Після цього контролер надає перегляд. Перегляд - це дизайн і вміст сторінки. Це кінцевий продукт, який буде повернуто користувачу.

Навколо Ruby on Rails утворилася велика екосистема додаткових плагінів з відкритим вихідним кодом («джеми», gems), які реалізують найбільш затребувані функції. «Джеми» бувають дуже різні: від низькорівневих, що відповідають за якийсь аспект внутрішньої роботи програми, до

високорівневих, що представляють собою окремі модулі для вирішення цілого спектру бізнес-задач. Це і стало причиною високої популярності фреймворка – можливість вибірково підключати окремі компоненти і бібліотеки дуже сильно прискорює розробку, а той факт, що використовувані рішення добре пристосовані і відлагоджуються роками, забезпечує надійність рішень, розроблених за допомогою таких підходів [21].

Саме з вибору джемів розпочинається розробка веб-додатку. Для розробки сайту Translation використовуються наступні джеми:

- friendly_id – створює URL-посилання у вигляді рядків змість числових ідентифікаторів;
- devise – це гнучке рішення для аутентифікації;
- babosa – транслітерація кирилі;
- activeadmin – створення адмінки;
- bootstrap – стилізація веб-сайту;
- jquery-rails – підключення бібліотеки jquery, яка йде разом з bootsrtap;
- will_paginate-bootstrap4 – створення пагінації на сайті;
- carrierwave – завантаження файлів;
- pg – СУБД PostgreSQL.

PostgreSQL – реляційна база даних. Це місце, де зберігається основна інформація веб-додатку. Ruby On Rails дозволяє нам не писати складні і нудні запити, а на основі Active Record писати запити дуже швидко. Active Record це М в MVC – модель, яка є шаром в системі, відповідальним за подання бізнес-логіки і даних. Active Record спрощує створення і використання бізнес-об'єктів, дані яких вимагають персистентного зберігання в базі даних. Сама по собі ця реалізація патерну Active Record є описом системи ORM (Object Relational Mapping). Object Relational Mapping (об'єктно-реляційне відображення) – це техніка, що з'єднує складні об'єкти додатка з таблицями в системі керування базами даних [17].

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№. д о к у м.	П і д п	Д а		40

За допомогою гему devise можна створити модель користувачів User і використовувати її для роботи з базою даних. Код для реалізації моделі користувачів наведено нижче:

```
class User < ApplicationRecord
  # Include default devise modules. Others available are:
  # :confirmable, :lockable, :timeoutable, :trackable and
  :omniauthable
  devise :database_authenticatable, :registerable,
         :recoverable, :rememberable, :validatable
  has_many :comments, dependent: :destroy
  validates :username, presence: true
  validates_uniqueness_of :username
end
```

Джем devise забезпечує автоматичне створення контролера Users. Для перевизначення або створення нових функцій потрібно користуватися основним контролером Application Controller.

Оскільки основною задачею сайту є прийняття замовлення на переклад, модель і контролер допоможуть це реалізувати. Наведемо деякі методи контролера замовлень Orders:

```
def index
  @orders = Order.all.order("created_at ASC")
end
def show
end
def new
  @order = Order.new
end
```

Модель Order має наступний вигляд:

```
class Order < ApplicationRecord
  mount_uploader :attachment, AttachmentUploader
  validates :name, :phone, :email, :notes, presence: true
  validates :email, uniqueness: true, on: :create
  validates :email, format: { with: URI::MailTo::EMAIL_REGEXP
  }
end
```

Стрічка “mount_uploader :attachment, AttachmentUploader” дозволяє завантажувати документ через джем carrierwave.

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		41

Дуже важливою частиною в веб-додатках на RoR є роутинг. Роутер Rails розпізнає URL і направляє його в екшен контролера або в додаток Rack. Він також може генерувати шляхи і URL, уникаючи необхідність жорстко прописувати рядки в виглядах. Весь код файлу routes.rb наведено нижче:

```
Rails.application.routes.draw do
  resources :orders, only: [:create]
  devise_for :admin_users, ActiveAdmin::Devise.config
  ActiveAdmin.routes(self)
  devise_for :users
  resources :posts do
    resources :comments
  end
  resources :tags, only: [:show]
  root 'static_pages#home'
  get 'home', to:'static_pages#home'
  get 'about', to:'static_pages#about'
  get 'services', to:'static_pages#services'
  get 'contacts', to:'static_pages#contacts'
  # For details on the DSL available within this file, see
  http://guides.rubyonrails.org/routing.html
end
```

Роутинг root дозволяє встановити сторінку, яка буде загрузатися як тільки користувач перейдете на веб-сайт. Роутинг resource дозволяє швидко оголошувати всі загальні маршрути для заданого ресурсного контролера. Замість оголошення окремих маршрутів для екшенів index, show, new, edit, create, update, destroy, ресурсний маршрут оголошує їх одним рядком коду. Ці екшени утворюють систему CRUD [18].

Варто зазначити, що найголовнішою частиною веб-сайту Translation є адміністративна панель, яка реалізована за допомогою джему active_admin. Для того, щоб перейти в адміністративну панель потрібно перейти за посиланням <https://translation/admin/>. Після цього користувач повинен пройти авторизацію. Тільки розробник сайту може зареєструвати адміністратора. Це забезпечує цілісність даних на веб-сайті.

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		42

Рисунок 3.12 - Форма входу в адміністративну панель

Адміністратор має доступ до всіх замовлення і блогу. Це означає, що він може створювати, видаляти і редагувати замовлення або статті (див. рисунок 3.13).

Id	Name	Phone	Email	From Lang	To Lang	Notes	Created At	Updated At	Attachment
11	Petro	54545	ipolon@ukr.net	Ukrainian	English	asdasd	May 28, 2019 00:34	May 28, 2019 00:34	/uploads/order/attachment/11/Untitled.pdf
10	Petro	651465265	bblab@ukr.net	Ukrainian	English	Bblabiba	May 27, 2019 23:30	May 27, 2019 23:30	/uploads/order/attachment/10/lab4_ts.docx
9	Ivan	5145646	slava@ukr.net	English	Ukrainian	stsdsd	May 27, 2019 23:16	May 27, 2019 23:16	/uploads/order/attachment/9/lab4_ts.docx
8	Ivan	6546465	slav@ukr.net	Ukrainian	English	Hello	May 27, 2019 22:29	May 27, 2019 22:29	/uploads/order/attachment/8/Untitled.pdf

Рисунок 3.13 - Вигляд адміністративної панелі

3.3 Тестування та верифікація

Тестування програмного забезпечення визначається як активність для перевірки відповідності фактичних результатів до очікуваних результатів і забезпечення того, щоб система програмного забезпечення не була дефектом.

Тестування програмного забезпечення також допомагає виявити помилки, прогалини або відсутні вимоги всупереч реальним вимогам.

Це можна зробити або вручну, або за допомогою автоматизованих інструментів.

Є також два поняття зв'язаних з тестуванням. Це верифікація і валідація. Верифікація програмного забезпечення – більш загальне поняття, ніж тестування. Метою верифікації є досягнення гарантії того, що верифікований об'єкт (вимоги або програмний код) відповідає вимогам, реалізований без непередбачених функцій. Процес верифікації включає в себе інспекції, тестування коду, аналіз результатів тестування, формування та аналіз звітів про проблеми. Валідація програмної системи - процес, метою якого є доказ того, що в результаті розробки системи ми досягли тих цілей, які планували досягти завдяки її використанню. Іншими словами, валідація – це перевірка відповідності системи очікуванням замовника.

На веб-сайті Translations користувачі можуть авторизуватися. Для успішної реєстрації або входу на веб-сайт відбуваються деякі перевірки. Прикладом таким перевірок може бути модель користувачів User. На рисунку 3.14 зображено реєстрацію користувачів.

Реєстрація

3 errors prohibited this user from being saved:

- Email can't be blank
- Password can't be blank
- Username can't be blank

Ім'я

E-mail

Пароль

(6 characters minimum)

Підтвердження паролю

Зареєструватися

Увійти

Рисунок 3.14 – Форма реєстрації

З рисунку 3.14 можемо побачити, що поля ім'я, емейл і пароль є обов'язковими і без них не відбудеться реєстрація користувача. Поле для

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		44

емейлу є унікальним для кожного користувача. Коли користувач при реєстрації використовує емейл, який вже існує, йому буде виводиться повідомлення про це.

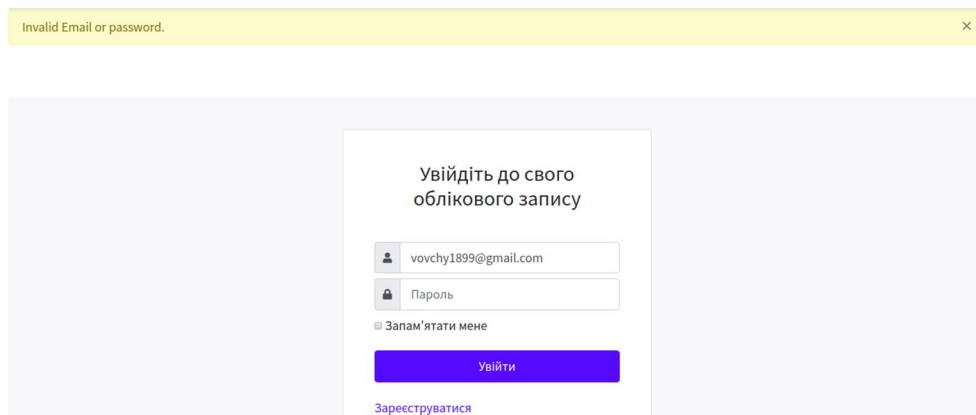


Рисунок 3.15 – Форма входу

Якщо авторизація пройшла успішно, користувач може залишати коментарі. Варто зазначити, що коментарі можуть бути видалені тільки його власником. Дану реалізацію зображено на рисунку 3.16.

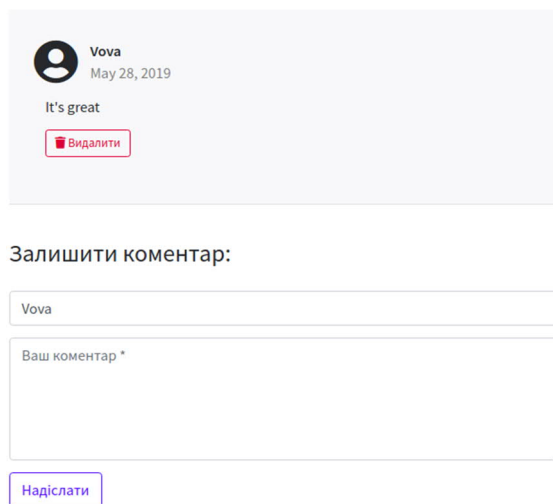


Рисунок 3.16 – Створення коментарів

Також така перевірка відбувається над формою для замовлення перекладу. Нижче наведено код моделі Order:

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№. д о к у м.	П і д п	Д а		45

```

class Order < ApplicationRecord
  mount_uploader :attachment, AttachmentUploader
  validates :name, :phone, :email, :notes, :attachment,
  presence: true
  validates :email, uniqueness: true, on: :create
  validates :email, format: { with: URI::MailTo::EMAIL_REGEX }
end

```

Метод `validates` відповідає за валідацію. Параметр `uniqueness` вказує на те, що об'єкт на якому здійснюється валідація повинен бути унікальним. Параметр `presence: true` вказує, що дані поля є обов'язковими для ведення.

Рисунок 3.17 – Форма замовлення

Якщо поля не проходять валідацію, користувачу приходить повідомлення про помилку. Як видно з рисунку 3.17, що поле для завантаження документа проходить валідацію на розширення файлу. Це означає, що користувач не може надіслати такі файли, як музику, фільми. Повідомлення вказує користувачу, які поля пройшли валідацію, а які ні. Тестування, валідація і верифікація є важливими етапами не тільки для самих користувачів, але для розробників веб-сайтів, що дозволяє їм уникати проблем з переповненням баз даних.

					Б Р . К С М . 07113/15.00.00.000 . П З	А р
З м	А р	№. д о к у м.	П і д п	Д а		46

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ КОМП'ЮТЕРНОЇ СИСТЕМИ

В цьому розділі дипломного проекту проводиться техніко-економічне обґрунтування розробки програмного забезпечення, а саме веб-сайту. Зокрема, здійснюється розрахунок витрат на розробку програмного забезпечення, експлуатаційних витрат, ціни споживання програмного забезпечення. В заключній частині визначаються показники економічної ефективності нового програмного продукту, обґрунтовуються відповідні висновки.

Розроблене програмне забезпечення призначене для системи автоматизованого документообігу перекладацького агентства.

4.1 Розрахунок витрат на розробку програмного забезпечення

Витрати на розробку і впровадження програмних засобів (K) включають:

$$K = K_1 + K_2, \quad (4.1)$$

де K_1 – витрати на розробку програмних засобів, грн;

K_2 – витрати на відлагодження і досліду експлуатацію програми рішення задачі на комп'ютері, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників ($B_{оп}$);
- витрати на відрахування у спеціальні державні фонди (B_{ϕ});
- витрати на покупні вироби ($B_{пв}$);

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		47

- витрати на придбання спецобладнання для проведення експериментальних робіт (O_B);
- накладні витрати (H);
- інші витрати (I).

4.1.1 Розрахунок витрат на оплату праці

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудомісткості відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці програмного забезпечення задіяні наступні спеціалісти - розробники, а саме – керівник проекту, студент-дипломник, консультант техніко-економічного розділу.

Таблиця 4.1 – Вихідні дані для розрахунку витрат на оплату праці

Посада виконавців	Місячний оклад, грн
Керівник ВКР, викладач	5950
Консультант техніко-економічного розділу, доцент	7293
Студент	1400

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{оп} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij}, \quad (4.2)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду, грн.

Середньо годинна ставка працівника може бути розрахована за формулою:

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PЧ_i}, \quad (4.3)$$

де C_{ij} – основна місячна заробітна плата розробника i -ої спеціальності j -го тарифного розряду, грн;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$PЧ_i$ – місячний фонд робочого часу працівника i -ої спеціальності j -го тарифного розряду, год (приймаємо 168 год).

Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 – Розрахунок витрат на оплату праці

Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год	Витрати на розробку, грн
Керівник ВКР, доцент	16	35,4	566,4
Консультант техніко-економічного розділу, доцент	2	63,8	127,6
Студент	120	8,3	996
Разом		1690	

4.1.2 Відрахування на соціальні заходи

Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5 % від суми заробітної плати:

$$B_{\phi} = \frac{20,5}{100} \cdot 1690 = 346,45 \text{ (грн)}. \quad (4.4)$$

4.1.3 Розрахунок витрат на матеріали та комплектуючі

У таблиці 4.3 наведений перелік купованих виробів і розраховані витрати на них.

Таблиця 4.3 – Розрахунок витрат на матеріали та комплектуючі

Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
Папір (формат А4)	уп	90,0	1	90,00	9,0	99,0
Ручка кулькова	шт	15,0	2	30,00	3,00	33,0
Олівець простий	шт	5,0	2	10,00	1,0	11,0
Зошит, 18 арк	шт	5,50	1	5,50	0,55	6,50
Тонер для принтера	уп	50	1	50	5,0	55,0
Разом						204,05

4.1.4 Витрати на використання комп'ютерної техніки

Витрати на використання комп'ютерної техніки включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№. д о к у м.	П і д п	Д а		50

забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ТНЕУ для комп'ютера типу IBM PC/ATX вартість години роботи становить 5,2 грн. Середній щоденний час роботи на комп'ютері – 2 години. Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4 – Розрахунок витрат на використання комп'ютерної техніки

Назва етапів робіт, при виконанні яких використовується комп'ютер	Час використання комп'ютера, год.	Витрати на використання комп'ютера, грн.
Проведення досліджень та оформлення їх результатів	75	390
Оформлення техніко-економічного розділу	10	36
Оформлення ВКР	15	54
Разом	100	360

4.1.5 Накладні витрати

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H = 1,5 \cdot 1690 = 2536 \text{ (грн)}. \quad (4.5)$$

4.1.6 Інші витрати

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I = 1690 \cdot 0,1 = 169 \text{ (грн)}. \quad (4.6)$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{OP} + B_{\Phi} + B_{ПВ} + H + I, \quad (4.7)$$

$$K_1 = 1690 + 346,45 + 204,5 + 2535 + 169 = 4944,95 \text{ (грн)}. \quad (4.8)$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{м.г.} \cdot t_{від}, \quad (4.9)$$

де $S_{м.г.}$ – вартість однієї машино-години роботи ПК, грн/год.

$t_{від}$ – комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 40 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 5,2 грн. Тому:

$$K_2 = 5,2 \cdot 100 = 520 \text{ (грн)} \quad (4.10)$$

На основі отриманих даних складаємо кошторис витрат на розробку програмного забезпечення (таблиця 2.5).

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
						52
З м	А р	№ д о к у м.	П і д п	Д а		

Таблиця 4.5 – Кошторис витрат на розробку програмного забезпечення

Найменування витрат	Сума витрат, грн
Витрати на оплату праці	1690
Відрахування у спеціальні державні фонди	346,45
Витрати на куповані вироби	204,5
Накладні витрати	2535
Інші витрати	169
Витрати на відлагодження і дослідну експлуатацію програмного продукту	520
Разом	5464,95

4.2 Визначення експлуатаційних витрат

Для оцінки економічної ефективності розроблюваного програмного продукту слід порівняти його з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення.

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

$$E_{\Pi} = E_{1\Pi} + E_{2\Pi}, \quad (4.11)$$

де E_{Π} – одноразові експлуатаційні витрати на ПЗ (аналог), грн.;

$E_{1\Pi}$ – вартість підготовки даних для експлуатації ПЗ (аналог), грн.;

$E_{2\Pi}$ – вартість роботи комп'ютера для розробки програмного забезпечення (аналог), грн.

Річні експлуатаційні витрати $B_{E\Pi}$ визначаються за формулою:

$$B_{E\Pi} = E_{\Pi} * N_{\Pi}, \quad (4.12)$$

де $N_{\text{п}}$ – періодичність експлуатації ПЗ (аналогу), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{\text{ІП}} = \sum_{i=1}^n n_i t_i c_i, \quad (4.13)$$

де i – категорії працівників, які приймають участь у підготовці даних

n_i – кількість працівників i -ої категорії, осіб;

t_i – трудомісткість роботи співробітників i -ої категорії по підготовці даних, год.;

c_i – середнього годинна ставка працівника i -ої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення:

$$c_i = \frac{c_i^0 (1+b)}{m}, \quad (4.14)$$

де c_i^0 – основна місячна заробітна плата працівника i -ої категорії, грн;

b – коефіцієнт, який враховує додаткову заробітну плату;

m – кількість робочих годин у місяці, год.

Для роботи з даними як для поточного програмного забезпечення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає: $c^0 = 1400$ грн. Тоді:

$$c_1 = \frac{1400}{22 * 8} = 7 \text{ (грн/год)}. \quad (4.15)$$

Трудомісткість підготовки даних для програмного забезпечення складає 1 год., для аналога 1,5 год.

					Б Р . К С М . 07113/15.00.00.000 . П З	А р
З м	А р	№ . д о к у м .	П і д п	Д а		54

Таблиця 4.6 – Розрахунок витрат на підготовку даних та реалізацію програмного забезпечення на комп'ютері

Час роботи співробітників, год	Середньогодинна заробітна плата, грн/год	Витрати, грн
Проектне рішення		
1	7	7
Аналог		
1,5	7	18,75

Витрати на експлуатацію комп'ютера визначається за формулою:

$$E_{2П} = t * S_{МГ} \quad (4.16)$$

де t – витрати машинного часу для реалізації програмного продукту, год;

$S_{МГ}$ – вартість однієї години роботи комп'ютера, грн/год.

$$E_{2П} = 1 \cdot 5,2 = 5,2 \text{ (грн)}; \quad E_{2А} = 1,5 \cdot 5,2 = 7,8 \text{ (грн)}; \quad (4.17)$$

$$E_{П} = 12,5 + 5,2 = 17,7 \text{ (грн)}; \quad E_{А} = 18,75 + 7,8 = 26,55 \text{ (грн)}; \quad (4.18)$$

$$B_{ЕП} = 17,7 \cdot 252 = 4460,4 \text{ (грн)}; \quad B_{ЕА} = 26,55 \cdot 252 = 6690,6 \text{ (грн)}. \quad (4.19)$$

4.3 Розрахунок ціни споживання програмного продукту

Ціна споживання – це витрати на придбання і експлуатацію програмного продукту за весь строк його служби:

$$Ц_{С(П)} = Ц_{П} + B_{(Е)NPV}, \quad (4.20)$$

де $Ц_{П}$ – ціна придбання програмного продукту, грн.

					Б Р . К С М . 07113/15.00.00.000 . П 3	А в
З м	А в	№ . д о к у м .	П і д п	Д а		55

$$Ц_{\Pi} = K \left(1 + \frac{\Pi_p}{100}\right) + K_0 + K_{\kappa}, \quad (4.21)$$

де K – кошторисна вартість;

Π_p – рентабельність;

K_0 – витрати на прив'язку та освоєння програмного забезпечення на конкретному об'єкті, грн;

K_{κ} – витрати на доукомплектування технічних засобів на об'єкті, грн.

$$Ц_{\Delta} = 5464,95 \cdot (1 + 0,3) = 7104,4 \text{ (грн)}. \quad (4.22)$$

Вартість витрат на експлуатацію програмного забезпечення (за весь час його експлуатації), грн:

$$B_{\text{епрв}} = \sum_{t=0}^T \frac{B_{\text{еп}}}{(1 + R)^t}, \quad (4.23)$$

де $B_{\text{еп}}$ – річні експлуатаційні витрати, грн;

T – термін служби програмного забезпечення, років;

R – річна ставка проценту банку.

$$B_{\text{епрв}} = \sum_{t=1}^5 \frac{4460,4}{(1 + 0,08)^t} = 17845,3 \text{ (грн)}, \quad (4.24)$$

$$B_{\text{епрв}} = \sum_{t=1}^5 \frac{6690,6}{(1 + 0,08)^t} = 26767,9 \text{ (грн)}. \quad (4.25)$$

Тоді ціна споживання програмного забезпечення дорівнюватиме:

					Б Р . К С М . 07113/15.00.00.000 . П 3	А в
З м	А в	№. д о к у м.	П і д п	Д а		56

$$Ц_{СП} = 71404,4 + 17845,3 = 24949,7 \text{ (грн)}. \quad (4.26)$$

Аналогічно визначається ціна споживання для аналогу:

$$Ц_{СА} = 6000,0 + 26767,9 = 32767,9 \text{ (грн)}. \quad (4.27)$$

4.4 Визначення показників економічної ефективності

Економічний ефект в сфері розробки програмного продукту:

$$E_{ПР} = Ц_{П} - Ц_{А}, \quad (4.28)$$

$$E_{ПР} = 71104,4 - 6000,0 = 1104,4 \text{ (грн)}. \quad (4.29)$$

Річний економічний ефект в сфері експлуатації:

$$E_{КС} = B_{ЕА} - B_{ЕП}, \quad (4.30)$$

$$E_{КС} = 6690,6 - 4460,4 = 2230,2 \text{ (грн)}. \quad (4.31)$$

Додатковий економічний ефект у сфері експлуатації:

$$\Delta E_{екс} = \sum_{t=1}^T E_{екс} (1+R)^{T-t}, \quad (4.32)$$

$$\Delta E_{екс} = \sum_{t=1}^5 2230,2 * (1+0,08)^{5-t} = 1306885 \text{ (грн)}. \quad (4.33)$$

					Б Р . К С М . 07113/15.00.00.000. П 3	А в
З м	А в	№. д о к у м.	П і д п	Д а		57

Сумарний ефект складає:

$$E = E_{\text{пр}} + \Delta E_{\text{екс}} = 1104,4 + 13068,85 = 14173,25 \text{ (грн)}. \quad (4.34)$$

Таблиця 4.7 – Показники економічної ефективності програмного забезпечення

Найменування	Одиниці вимірювання	Значення показників	
		Базовий варіант	Новий варіант
Капітальні вкладення	грн.	–	5464,95
Ціна придбання	грн.	6000,0	7104,4
Річні експлуатаційні витрати	грн.	17845,3	26767,9
Ціна споживання	грн.	24949,7	32767,9
Економічний ефект в сфері проектування	грн.	–	1104,4
Економічний ефект в сфері експлуатації	грн.	–	2230,2
Додатковий ефект в сфері експлуатації	грн.	–	13068,85
Сумарний ефект	грн.	14173,25	

4.5 Висновки

В даному розділі проведено розрахунок витрат на розробку програмного забезпечення. Здійснено порівняння з існуючим аналогом, і цим показано, що вказане програмне забезпечення має переваги в порівнянні з аналогами, зокрема: надійність, простота використання, гнучкість, зручність. Згідно проведеного економічного обґрунтування зазначене програмне забезпечення є конкурентоздатним. Крім того, отримано економічний ефект у розмірі 14173,25 грн. і тому розробка і впровадження цього програмного забезпечення є економічно доцільними.

ВИСНОВКИ

В даній бакалаврській роботі було проаналізовано технології розробки веб-сайтів для подальшої розробки системи документообігу перекладацького агентства.

В роботі було розглянуто розвиток технологій розробки і функціонування веб-технологій та проаналізовано принципи їх побудови: як розробити макет та веб-дизайн сайту, як правильно розподілити доступ прав між користувачами, як проводити тестування та верифікацію.

Основною перевагою мови програмування Ruby і фреймворка Ruby on Rails вважається швидкість розробки. Практика показує, що швидкість розробки проектів на RoR збільшується на 30-40 відсотків по відношенню до будь-якої іншої мови програмування або фреймворку. В першу чергу приріст швидкості розробки визначається великим набором готових до роботи інструментів RoR, колосальним набором готових рішень в співтоваристві, мови Ruby і простоті програмування на ньому.

Після порівняння з іншими фреймворками можна сказати, що :

– у Ruby on Rails є можливість гнучко налаштувати ваш роутинг, вид адрес, назви основних розділів. Є можливість швидко змінити адресу в одному місці без необхідності зміни цієї адреси в усьому проекті. У співтоваристві RoR-розробників активно використовується ідеологія REST. Адреси сторінок в проектах на Ruby on Rails завжди зрозумілі, красиві, чудово розуміються пошуковими системами;

– у Ruby on Rails прекрасно реалізовані інструменти, що дозволяють перевіряти вхідні дані. Користувачі заповнюють форми і потрібно перевірити правильність введення адреси електронної пошти, наявність пароля або необхідну мінімальну довжину логіна - засоби Rails вам в цьому допоможуть;

– повсякденна проблема багатьох проектів на PHP – неможливість зрозумілими засобами та інструментами контролювати структуру бази даних.

					Б Р . К С М . 07113/15.00.00.000 . П 3	А р
З м	А р	№ . д о к у м .	П і д п	Д а		59

Зміни в структуру часто вносяться вручну і прямо в базу. Через це в проекті з'являються численні незрозумілі поля і таблиці, про які вже ніхто нічого не пам'ятає. В Ruby on Rails існують інструменти роботи з базами даних – «міграції». Структура бази даних зберігається в коді програми і конфігурується з проекту;

– Ruby on Rails за замовчуванням сильно орієнтовані на безпеку проекту. При використанні інструментів RoR виключені SQL ін'єкції і XSS атаки. Всі вхідні параметри екрануються за замовчуванням. Виведені змінні в шаблонах також екрануються, тільки якщо ви не вказали зворотної опції.

В роботі було обрано реалізацію веб-сайту призначеного для перекладу документів на основі вже існуючих сайтів. В результаті виконання практичної частини було розроблено власний веб-дизайн, додано адміністративну панель, авторизацію користувачів, завантаження документів і створено блог.

Розроблена автоматизована система документообігу перекладацького агентства має практичне значення, що підтверджено довідкою про впровадження (додаток В).

Результати дипломного проектування доповідалися на інтернет-конференції «Науково-практична конференція інтелектуальні комп'ютерні системи та мережі» (додаток Б).

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№. д о к у м.	П і д п	Д а		60

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жуков І.А., Гуменюк В.О., Альтман І.Є. Комп'ютерні мережі та технології. НАУ, 2004. 276 с.
2. Антонов В.М. Комп'ютерні мережі та технології. «МК – Прес», 2005. 480 с.
3. Структура сайту: веб-сайт. URL: <https://internet-marketings.ru/struktura-sajta-razrabotka-v-vide-sxemy/> (дата звернення: 12.03.2019).
4. Эд Титтел, Джефф Ноубл. HTML, XHTML и CSS для чайников. 2011. 400 с.
5. Backend Web Frameworks: веб-сайт. URL: <https://hackernoon.com/7-best-web-development-backend-frameworks-in-2018-22a5e276cdd> (дата звернення: 12.04.2019).
6. HTML&CSS: веб-сайт. URL: <https://varvy.com/pagespeed/display.html> (дата звернення: 27.03.2019)
7. Итан Маркотт, Иванов и Фербер. Отзывчивый веб-сайт . 2012. 176 с.
8. Петлюшкин А.В. HTML в Web-дизайне. Петербург, 2004. 400 с.
9. HTML: веб-сайт. URL: <http://programming.in.ua/web-design/html/73-html-introduction.html> (дата звернення: 03.08.2019).
10. Web Frameworks: веб-сайт. URL: <https://djangostars.com/blog/what-is-a-web-framework/> (дата звернення: 16.05.2019).
11. The Difference Between Frontend And Backend Web Development: веб-сайт. URL: <https://careerfoundry.com/en/blog/web-development/whats-the-difference-between-frontend-and-backend/> (дата звернення: 29.03.2019).
12. Websites work: веб-сайт. URL: <https://tinyurl.com/y4ezs5oc> (дата звернення: 16.05.2019).
13. Domain: веб-сайт. URL: <https://hostiq.ua/ukr/info/what-is-domain/> (дата звернення: 18.05.2019).

					Б Р . К С М . 07113/15.00.00.000. П З	А д
З м	А д	№. д о к у м.	П і д п	Д а		61

14. What is HTTP?: веб-сайт. URL: <https://searchwindevelopment.techtarget.com/definition/HTTP> (дата звернення: 24.05.2019)

15. HTTP: веб-сайт. URL: <https://www.cloudflare.com/learning/ssl/what-is-https/> (дата звернення: 25.05.2019).

16. Тейт Б., Хиббс К. Ruby on Rails. Быстрая веб-разработка. Петербург, 2008. 224 с.

17. Хэнссон Д. Х., Томас Д. Гибкая разработка веб-приложений в среде Rails. Питер, 2008. 720 с.

18. Фоулер Ч. Rails. Сборник рецептов. Питер, 2007. 256 с.

19. Фернандес О. Путь Rails. Подробное руководство по созданию приложений в среде Ruby on Rails. Символ-Плюс, 2008. 768 с.

20. Руби С., Томас Д., Хэнссон Д. Х. Гибкая разработка веб-приложений в среде Rails. Питер, 2012. 464 с.

21. Hartl M. Ruby on Rails Tutorial. Addison-Wesley, 2015. 664 с.

22. Фултон Х. Программирование на языке Ruby . Пресс, 2007. 688 с.

23. Методичні вказівки до написання техніко-економічного розділу дипломних проектів освітньо-кваліфікаційного рівня «бакалавр» підготовки 6.050102 комп'ютерна інженерія/ І.Р. Паздрій – Тернопіль: ТАНГ, 2014.– 37 с.

24. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня “Бакалавр” напряму підготовки 6.050102 «Комп'ютерна інженерія» фахового спрямування «Комп'ютерні системи та мережі» / О.М. Березький, Л.О.Дубчак, Г.М. Мельник, Ю.М. Батько, С.В. Івасьєв / Під ред. О.М. Березького. - Тернопіль: ТНЕУ, 2016.–65с.

					Б Р . К С М . 07113/15.00.00.000. П 3	А р
З м	А р	№ д о к у м .	П і д п	Д а		62