ЗМІСТ

[ВСТУП 3](#_Toc9299498)

[1 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ 4](#_Toc9299499)

[1.1. Аналіз предметної області 5](#_Toc9299500)

[1.2. Огляд існуючих програмних продуктів 5](#_Toc9299501)

[1.2.1. «JustNotepad» 5](#_Toc9299502)

[1.2.2. «Напишите заметку» 6](#_Toc9299503)

[1.2.3. «Pnote.ru» 7](#_Toc9299504)

[1.2.4. «Notes.io» 8](#_Toc9299505)

[1.3. Постановка задачі 9](#_Toc9299506)

[2 СПЕЦИФІКАЦІЯ ВИМОГ ДО ВЕБ-ДОДАТКУ 11](#_Toc9299507)

[2.1. Загальна інформація щодо веб-застосунку 11](#_Toc9299508)

[2.1.1 Огляд продукту 11](#_Toc9299509)

[2.1.2 Мета 12](#_Toc9299510)

[2.1.3 Межі 12](#_Toc9299511)

[2.1.4 Масштаб проекту 12](#_Toc9299512)

[2.1.5 Обмеження реалізації 13](#_Toc9299513)

[2.1.6 Перспективи продукту 13](#_Toc9299514)

[2.1.7 Діаграми потоків даних 13](#_Toc9299515)

[2.2 Вимоги до веб-додатку 14](#_Toc9299516)

[2.2.1 Функціональні вимоги 14](#_Toc9299517)

[2.2.2 Нефункціональні вимоги 15](#_Toc9299518)

[2.2.2.1 Надійність 15](#_Toc9299519)

[2.2.2.2 Доступність 15](#_Toc9299520)

[2.2.2.3 Безпека 15](#_Toc9299521)

[2.2.2.4 Супровід 15](#_Toc9299522)

[2.2.2.5 Кросплатформність 16](#_Toc9299523)

[2.2.2.6 Продуктивність 16](#_Toc9299524)

[2.3 Опис обраних технологій для розробки веб-додатку 16](#_Toc9299525)

[2.3.1 HTML5 16](#_Toc9299526)

[2.3.2 CSS3 17](#_Toc9299527)

[2.3.3 ADO.NET 19](#_Toc9299528)

[2.3.4 СУБД MS SQL Server 20](#_Toc9299529)

[2.3.5 Microsoft Visual Studio 22](#_Toc9299530)

[2.4 Опис архітектури веб-додатку 24](#_Toc9299531)

[2.4.1 Діаграма розміщення 24](#_Toc9299532)

[2.4.2 Діаграма компонентів 25](#_Toc9299533)

[2.5 Модель бази даних 26](#_Toc9299534)

[3 ОПИС ВЕБ-ДОДАТКУ 27](#_Toc9299535)

[3.1 Тестування веб-додатку 27](#_Toc9299536)

[3.1.1 Верифікація та валідація веб-додатку 27](#_Toc9299537)

[3.1.2 Ручне тестування 28](#_Toc9299538)

[3.2 Керівництво користувача 30](#_Toc9299539)

[3.2.1 Загальний опис можливостей 30](#_Toc9299540)

[3.2.2 Об’єкти 30](#_Toc9299541)

[ВИСНОВКИ 33](#_Toc9299542)

[СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ 34](#_Toc9299543)

# ВСТУП

21 століття - століття технологій, яке з кожним роком обростає новими впровадженнями в цю велику частину нових знань. Вплив нових методів роботи з обчислювальною технікою зростає з кожним роком в експоненційному напрямку. З кожним роком значення інформаційних технологій в нашому повсякденному житті зростає. Для інформаційного суспільства характерно повне задоволення інформаційних потреб населення при завершенні формування єдиного інформаційного середовища, що визначає нову культуру як суспільства в цілому, так і кожної людини окремо. Все менше і менше люди користуються фізичними ресурсами для зберігання даних, в тому числі, і текстових. Для цього стає розумним використовувати онлайн блокнот в всесвітній мережі.

Онлайн блокнот має велику перевагу через можливості швидкої доступності та коригування даних в будь-який момент. Метою курсової роботи є розробка веб-додатки, яке дозволить створювати і зберігати текстову інформацію і фотографії. При виконанні даної роботи буде вивчена предметна область і різні аспекти веб-додатки, в тому числі, аналіз своєрідності роботи блокнотів, який допоможе розробити сучасне веб-додаток з використанням актуальних технологій.

1. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ
   1. **Аналіз предметної області**

21 століття - століття технологій, яке з кожним роком обростає новими впровадженнями в цю велику частину нових знань. Вплив нових методів роботи з обчислювальною технікою зростає з кожним роком в експоненційному напрямку. З кожним роком значення інформаційних технологій в нашому повсякденному житті зростає. Все менше і менше люди користуються матеріальними ресурсами для зберігання даних, в тому числі, і текстових. Для цього стає розумним використовувати онлайн блокнотів в всесвітній мережі.

Онлайн блокнот має велику перевагу через можливості швидкої доступності та коригування даних в будь-який момент. Метою курсової роботи є розробка веб-додатку, який дозволить створювати і зберігати текстову інформацію і фотографії. При виконанні даної роботи буде вивчена предметна область і різні аспекти веб-додатку, в тому числі, аналіз своєрідності роботи блокнотів, який допоможе розробити сучасне веб-додаток з використанням актуальних технологій.

* 1. **Огляд існуючих програмних продуктів**

Огляд існуючих онлайн нотатників, спрямованих на зберігання тієї чи іншої текстової інформації, є дуже важливим пунктом в цій роботі. Мета цього – порівняти всі обрані блокноти, виявити їх недоліки і переваги.

* + 1. **«JustNotepad»**

JustNotepad – безкоштовний нотатник для швидких заміток [1].

Інтерфейс нотатника зображений на рисунку 1.1.

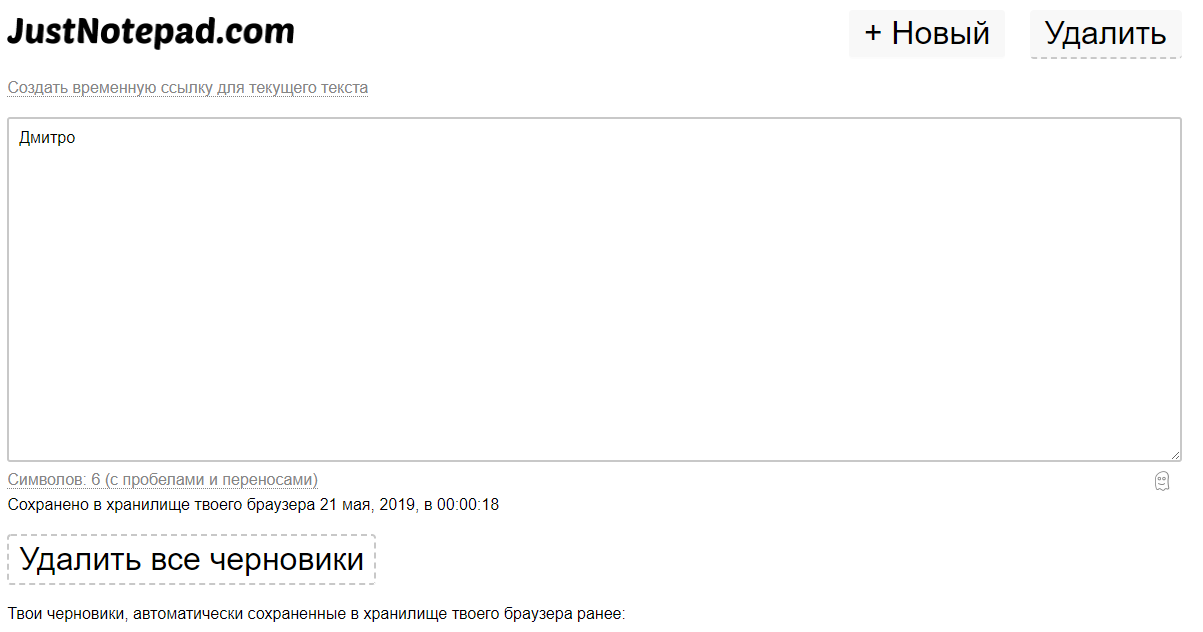


Рисунок 1.1 – Інтерфейс нотатника «JustNotepad»

Блокнот має наступні особливості:

* створення нового запису;
* видалення запису;
* зберігання чернетки в кеш браузеру протягом доби;
* видалення всіх чернеток одночасно;
* підрахунок кількості символів.
  + 1. **«Напишите заметку»**

Блокнот розроблений для тих, хто хоче швидко занести та зберегти нотатку в інтернеті. [2].

Інтерфейс додатку зображений на рисунку 1.2.

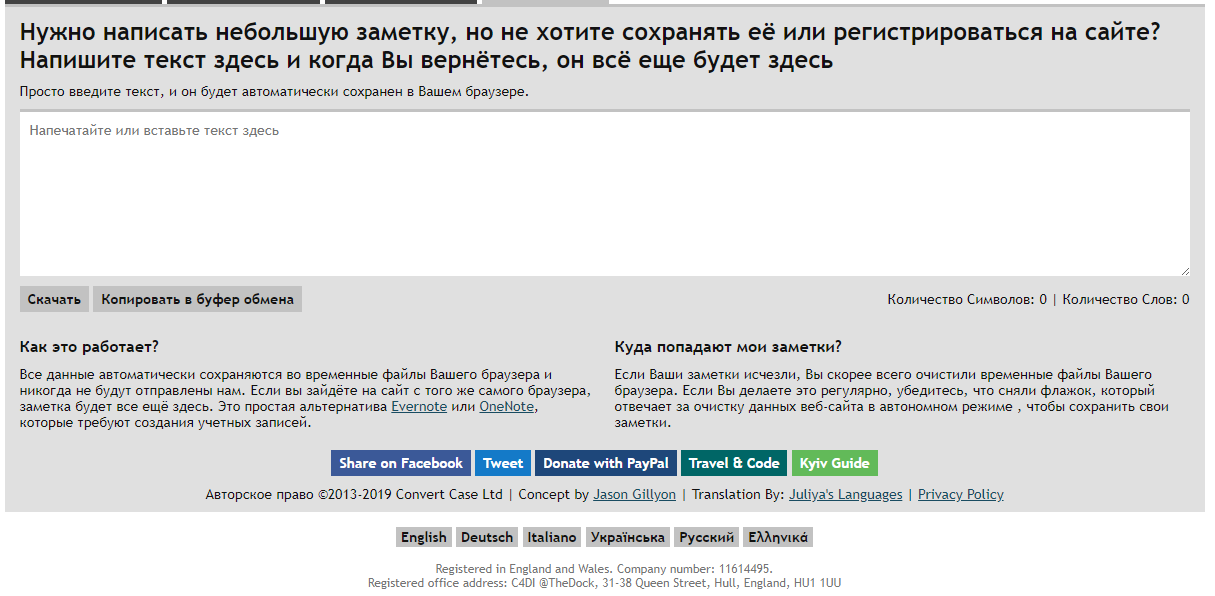


Рисунок 1.2 – Інтерфейс «Напишите заметку»

Нотатник має наступні особливості:

* зберігання нотатки на ПК;
* створення нової нотатки;
* видалення запису;
* зберігання нотатки в кеш браузеру протягом доби;
* зміна мови інтерфейсу;
* підрахунок кількості символів.
  + 1. **«Pnote.ru»**

Онлайн блокнот розроблений для тих, кому набридло використовувати для записів ручку та паперовий записник [3].

Інтерфейс додатку зображений на рисунках 1.3 та 1.4.

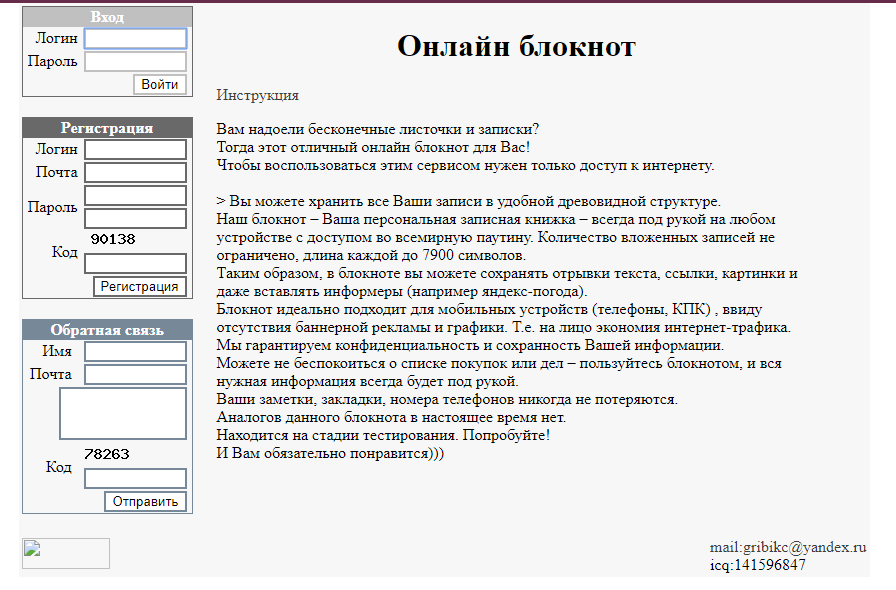


Рисунок 1.3 – Інтерфейс «Pnote.ru»

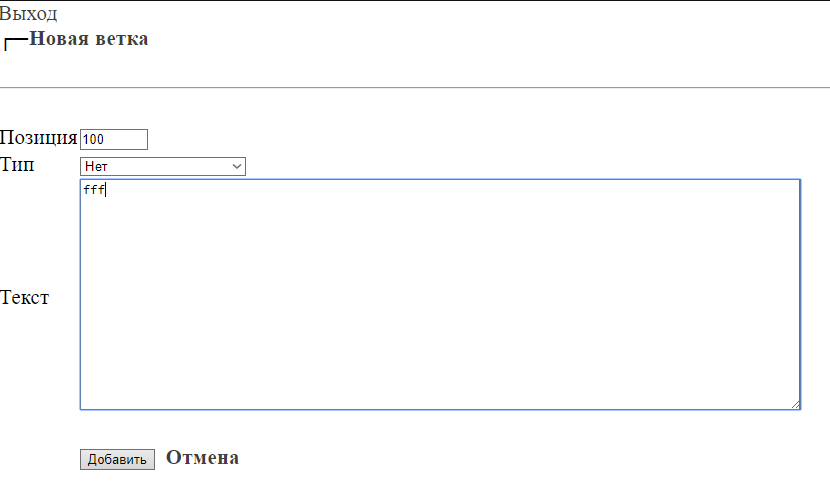


Рисунок 1.4 – Інтерфейс «Pnote.ru»

Блокнот має наступні особливості:

* реєстрація;
* створення нового запису;
* встановлення позиції для запису;
* зворотній зв’язок з розробником.
  + 1. **«Notes.io»**

Один з найкрасивіших, швидких і зручних онлайн блокнотів – Notes.io. З його допомогою можна створити запис довжиною в 18000 знаків за кілька секунд. [4].

Інтерфейс додатку зображений на рисунку 1.5.



Рисунок 1.5 – Інтерфейс нотатника «Notes.io»

Блокнот має наступні особливості:

* створення нового запису;
* приємний інтерфейс;
* можливість поділитися лінком на нотатку.

Таблиця 1.1 – Порівняння блокнотів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | JustNotepad | Напишите заметку | Pnote.ru | Notes.io |
| Дизайн | Простий, приємний для ока | Задовільний | Задовільний | Стильний, сучасний |
| Рівень функціональності | Середній | Середній | Високий | Низький |
| Можливість створювати нову замітку | + | + | + | + |
| Зберегти замітку | + | + | + | - |
| Підтримка в реальному часі | + | + | + | - |

Вище перераховані нотатники задовольняють мінімальним критеріям. Аналізуючи статистику відвідування онлайн записників, можна прийти до висновку, що на даний момент зберігання інформації в онлайн нотатниках є досить актуальним і популярним і втратити актуальність найближчим часом не зможуть.

* 1. **Постановка задачі**

Онлайн блокноти в даний час набирають велику популярність і багато користувачів віддають їм перевагу перед звичайним паперовим блокнотом. Онлайн блокнот має наступні переваги:

* доступність – використання в будь-який час доби і в будь-якому місці;
* можливість створювати нові записи і їх редагування;
* збереження введеної інформації і неможливість її втрати.
* можливість додати фотографію.

Завдання зберігання і обробки інформації є актуальним для сфери організації роботи онлайн блокноту.

Метою роботи є створення онлайн нотатника для декількох користувачів, що дозволить додавати нові записи, фотографії, редагувати і видаляти їх. Програмний продукт повинен бути доступним для користування людей різних вікових категорій, а саме бути простим і зрозумілим.

Необхідно забезпечити веб-додаток можливістю додавання нових записів, їх редагування, видалення й перегляду та прикріпленням фотографії до нотатку.

Відповідно до поставленої мети і вимогами були сформульовані наступні завдання:

* дослідження і опис предметної області – організація роботи онлайн блокноту;
* розробка моделі даних;
* вибір бази даних і засобів розробки веб-додатку;
* розробка веб-додатку;
* розробка простого і зручного інтерфейсу;
* створення можливості редагування, додавання, перегляду і видалення записів та фотографій;

Для досягнення поставлених завдань виділимо наступні задачі:

* вивчення сфери онлайн нотатників;
* дослідження структури додатку;
* аналіз програмних засобів створення блокнотів;
* вибір архітектури та розробка нотатника, використовуючи результати проведеного аналізу;
* тестування блокноту;
* обґрунтування ефективності даного проекту;

Об’єкт дослідження – онлайн блокнот.

Предмет дослідження – технології .NET для створення онлайн нотатника «Notepad».

1. СПЕЦИФІКАЦІЯ ВИМОГ ДО ВЕБ-ДОДАТКУ

## **2.1. Загальна інформація щодо веб-застосунку**

Онлайн нотатник «Notepad» призначений для надання можливості швидкого занесення та зберігання текстової інформації не використовуючи ручки та паперового блокноту. Даний записник дозволить користувачам додавати, переглядати, редагувати та видаляти нотатки та прикріпляти до них фотографії кожного дня та в будь-який час. Введені дані зберігаються в інформаційній базі даних, що дозволяє переглядати інформацію всім користувачам необмежену кількість раз.

* + 1. **Огляд продукту**

«Notepad» – це веб-додаток, який допоможе користувачу швидко занести інформацію в записник. Контекстна діаграма (рис 2.1) показує зовнішній об'єкт і системні інтерфейси для версії 1.0.

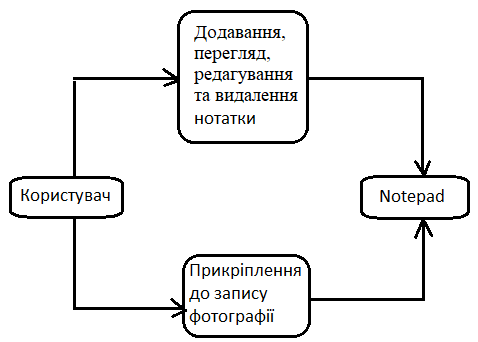


Рисунок 2.1 – Контекстна діаграма процесу внесення змін до нотатника

Передбачається випуск кількох версій системи для отримання універсального онлайн блокноту з усіма можливими функціями, який орієнтовано на зберігання не одного тільки тексту і фотографії, а на всі текстові файли з розширенням .doc, .docx, .txt і т.д.

* + 1. **Мета**

Визначити користувачів веб-додатку для покращення інтерфейсу та розширення функціональності.

* + 1. **Межі**

Використання веб-додатку поширюється на всі країни.

Веб-додаток застосовується для всіх користувачів.

* + 1. **Масштаб проекту**

Веб-додаток дозволить користувачам додавати нотатки з фотографією через інтернет для їх зберігання, перегляду та редагування.

Функції, реалізація яких запланована в цьому випуску:

1. Створення, перегляд, зміна і видалення нотатки.
2. Додавання та перегляд приєднаної до нотатки фотографії.
3. Забезпечення доступу до веб-додатку через мережу інтернет.
4. Збереження нових нотаток до бази даних.
   * 1. **Обмеження реалізації**

Веб-додаток створений за допомогою HTML 5.0, CSS3, мови C# та технології ADO.NET.

* + 1. **Перспективи продукту**

Досягти збільшення середнього рейтингу з щоквартального опитування за відповідністю потребам роботи веб-додатку на 0,7 балу протягом 8 місяців після першого випуску системи і на 1,0 бали протягом 12 місяців після другого випуску системи.

* + 1. **Діаграми потоків даних**

Для опису присутньої функціональності була побудована діаграма потоків даних (DFD – data flow diagram) (рис 2.2).

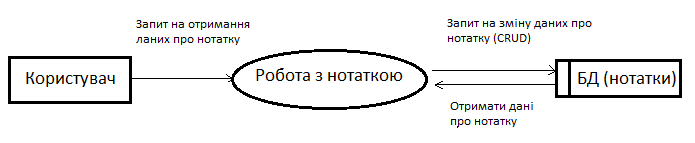


Рисунок 2.2 – Діаграма потоків даних (користувач)

Діаграми потоків даних описують зовнішні по відношенню до системи джерела і адресати даних, логічні функції, потоки даних та сховища даних, до яких здійснюється доступ.

Інформаційна система приймає ззовні потоки даних. Для позначення елементів середовища функціонування системи використовується поняття зовнішньої сутності. Для позначення елементів середовища функціонування системи використовується поняття зовнішньої сутності. Всередині системи існують процеси перетворення інформації, які породжують нові потоки даних. Потоки даних можуть надходити на вхід до інших процесів, розміщуватися (і вилучатись) в накопичувачі даних, передаватися до зовнішніх сутностей.

* 1. **Вимоги до веб-додатку**
     1. **Функціональні вимоги**

Функціональні вимоги це вимоги до програмного забезпечення, які описують внутрішню роботу системи, її поведінку; специфічні функції, які має виконувати система. На відміну від нефункціональних вимог, які визначають якою система повинна бути, функціональні вимоги визначають, що система повинна робити. Такі вимоги наведені в таблиці 2.1, де під об’єктом мається на увазі «нотатка». Так як додаток розрахований на одного користувача, тому опції вибору «об’єкта» немає.

Таблиця 2.1 – Опис функціональних вимог до веб-додатку

|  |  |
| --- | --- |
| Вимога | Описання |
| Об’єкт. Видалення | Для видалення нотатки, користувач має натиснути на кнопку «Удалить», попередньо вибравши рядок в таблиці, який буде підлягати видаленню. Після проведених дій, запис буде видалений зі списку нотаток. |
| Об’єкт. Редагування | Для редагування даних об’єкта, користувач в таблиці об’єктів повинен натиснути на кнопку «Редактировать», попередньо вибравши рядок в таблиці, який буде підлягати редагуванню, заповнити поля та натиснути на кнопку «Сохранить». Після редагування, запис буде змінено. |
| Об’єкт. Створити | Для створення об’єкта, користувач має в таблиці розділів натиснути на кнопку «Создать». У відкритому вікні ввести потрібну інформацію в поля вводу: «Название», «Текст», «Фотография». Для збереження об’єкта, користувач має натиснути на кнопку «Сохранить». Після проведених дій, об’єкт буде збережено нотатку до програми. |
| Об’єкт. Переглянути | Для перегляду об’єкта, користувач має в таблиці розділів натиснути на кнопку «Посмотреть», попередньо вибравши рядок в таблиці, який буде підлягати перегляду. Після проведених дій, запис буде виведений у вигляді таблиці. |

Функціональні вимоги до програмного забезпечення визначаються на першій стадії процесу розробки ПЗ — на етапі аналізу вимог.

* + 1. **Нефункціональні вимоги** 
       1. **Надійність**

Якщо з'єднання між користувачем і системою розривається до того, як дані були додані, змінені або видалені, система веб-додатку повинна дозволити користувачеві відновити дані для перегляду.

* + - 1. **Доступність**

Веб-додаток «Notepad» повинен бути доступним для всіх користувачів, які під’єднані до мережі інтернет. Веб-додаток працює 24 години на добу, тобто не має часових обмежень.

* + - 1. **Безпека**

Система повинна надавати користувачеві повний доступ до внесення змін в даних про нотатку.

* + - 1. **Супровід**

Читабельність коду можна розглядати як важливу складову супроводу. Код повинен бути зручним для аналізу, змін та тестування.

Модифікація, яка здійснюється для усунення дефектів, удосконалення або адаптації веб-додатку до змін в умовах функціонування, a також в складі і особливостях необхідних функцій є також одним із важливих факторів.

Додавання, видалення, зміна нотатки повинні бути зручними для користувача веб-додатку.

* + - 1. **Кросплатформенність**

Веб-додаток повинен мати властивість кросбраузерності.

Веб-додаток інсталяції не потребує, так як він буде доступний користувачеві за допомогою мережі інтернет через браузер.

* + - 1. **Продуктивність**

Веб-додаток повинен мати відмінну швидкість доступу до бази даних та завантаження сторінок, не навантажувати процесор та оперативну пам'ять.

* 1. **Опис обраних технологій для розробки веб-додатку**
     1. **HTML5**

HTML (від англ. HyperText Markup Language – «мова гіпертекстової розмітки») – стандартизована мова розмітки документів у Всесвітній павутині. Більшість веб-сторінок містять опис розмітки на мові HTML (або XHTML). Мова HTML інтерпретується браузерами; отриманий в результаті інтерпретації форматований текст відображається на екрані монітора комп'ютера або мобільного пристрою [5].

Мова HTML до 5-ї версії визначалася як додаток SGML (стандартної узагальненої мови розмітки за стандартом ISO 8879). Специфікації HTML5 формулюються в термінах DOM (об'єктній моделі документа).

Мова XHTML є більш суворим варіантом HTML, вона є наслідком синтаксису XML і є додатком мови XML в області розмітки гіпертексту.

У всесвітній павутині HTML-сторінки, як правило, передаються браузерам від сервера по протоколах HTTP або HTTPS, у вигляді простого тексту або з використанням шифрування.

Переваги HTML5:

* легкість у створенні програмного коду;
* відсутність таблиць div;
* використання елементів article, footer, nav, header та ін.;
* відтворення Flash (анімації, відеоролики, музику) у вікні браузера і для цього не потрібні драйвера і утиліти;
* обмін даними з сервером став поліпшеним;
* отримання оповіщень про помилки при роботі сайту.

Недоліки HTML:

* трудомісткість;
* дорога вартість;
* низька безпека;
* не всі браузери мають можливість правильного відображення сайту на HTML5.

В цілому дана технологія є непоганою заміною застарілого HTML.

* + 1. **CSS3**

CSS (англ. Cascading Style Sheets – каскадні таблиці стилів) – формальна мова опису зовнішнього виду документа, написаного з використанням мови розмітки [6].

Переважно використовується як засіб опису, оформлення зовнішнього вигляду веб-сторінок, написаних за допомогою мов розмітки HTML і XHTML, але може також застосовуватись до будь-яких XML-документів, наприклад, до SVG або XUL.

CSS використовується творцями веб-сторінок для задання кольорів, шрифтів, розташування окремих блоків і інших аспектів представлення зовнішнього вигляду веб-сторінок. Основною метою розробки CSS було розділення опису логічної структури веб-сторінки (яка проводиться за допомогою HTML або інших мов розмітки) від опису зовнішнього вигляду цієї веб-сторінки (яка тепер проводиться за допомогою формального мови CSS). Такий поділ може збільшити доступність документа, надати велику гнучкість і можливість управління його поданням, а також зменшити складність і повторюваність в структурному вмісті. Крім того, CSS дозволяє представляти один і той же документ в різних стилях або методах виведення, таких як екранне уявлення, друковане подання, читання голосом (спеціальним голосовим браузером або програмою читання з екрану), або при виведенні пристроями, що використовують шрифт Брайля.

Переваги CSS:

1. Простота самої мови CSS разом з принципом відділення оформлення від змісту дає можливість скоротити час на розробку і підтримку сайту.
2. Є кілька варіантів дизайну сторінки для перегляду на різних пристроях. Наприклад, дизайн на екрані комп'ютера розрахований на одну ширину, і буде повністю виводитися на екран, а на мобільних пристроях він буде підлаштовуватися до розмірів екрану і деякі елементи будуть виключені від показу, також і при друці, буде друкуватися потрібний текст, без зайвого (наприклад, без шапки меню).
3. Зменшується час завантаження сторінок web-сайту за рахунок перенесення правил представлення даних в окремий CSS-файл. Завдяки цьому браузер завантажує тільки структуру документа, а також дані, що зберігаються на сторінці, а представлення цих даних завантажується браузером тільки один раз і може бути закешованим, – завдяки цьому зменшується трафік, час завантаження, а також навантаження на сервер.
4. Простота зміни дизайну. Один CSS управляє відображенням безлічі HTML-сторінок. Коли виникає необхідність змінити дизайн сайту, то нема чого правити кожну сторінку. Для подальшої зміни дизайну всього лише потрібно змінити CSS-файл, і як результат, зміна дизайну робиться швидше.
5. CSS надає додаткові можливості форматування.
6. Підвищення сумісності з різними платформами за рахунок використання web-стандартів.

Недоліки CSS:

1. Різне відображення верстки в різних браузерах. Якщо браузери застарілі, то можливо, що одні й ті ж дані CSS по-різному ними інтерпретуються.
2. Необхідність виправляти не тільки один CSS-файл, але і теги HTML. Часто зустрічається на практиці необхідність виправляти не тільки один CSS-файл, але і теги HTML, які пов'язані з селекторами CSS. Іноді це значно збільшує час редагування, а також і тестування.

Отже, CSS – це найсильніший інструмент і один з основних складових практично будь-якого web-сайту. Без CSS просто неможливо уявити сучасну діяльність web-розробників. Сьогодні CSS – це загальноприйнятий стандарт розробки, який приймається усіма без винятку компаніями-розробниками, що явно показує його значимість і необхідність.

* + 1. **ADO.NET**

ASP.NET (Active Server Pages для .NET) – платформа розробки веб-додатків, до складу якої входить: веб-сервіси, програмна інфраструктура, модель програмування від компанії Майкрософт [7]. ASP.NET входить до складу платформи .NET Framework і є розвитком попередньої технології Microsoft ASP.

ASP.NET зовні багато в чому зберігає схожість з попередньою технологією ASP, що дозволяє розробникам відносно легко перейти на ASP.NET. У той же час внутрішній устрій ASP.NET істотно відрізняється від ASP, оскільки вона заснована на платформі .NET і, отже, використовує всі нові можливості, що надаються цією платформою.

Оскільки ASP.NET ґрунтується на Common Language Runtime (CLR), яка є основою всіх додатків Microsoft .NET, розробники можуть писати код для ASP.NET, використовуючи мови програмування, що входять в комплект .NET Framework (C #, Visual Basic.NET, J# і JScript .NET).

Програмна модель ASP.NET ґрунтується на протоколі HTTP і використовує його правила взаємодії між сервером і браузером. При формуванні сторінки закладена абстрактна програмна модель Web Forms і на ній заснована основна частина реалізації програмного коду.

Переваги ASP.NET перед ASP:

* компільований код виконується швидше, більшість помилок усуваються ще на стадії розробки;
* значно поліпшена обробка помилок під час виконання запущеної готової програми, з використанням блоків try..catch;
* призначені для користувача елементи управління (controls) дозволяють виділяти часто використовувані шаблони, такі як меню сайту;
* використання метафор, вже застосовуються в Windows-додатках, наприклад, таких як елементи управління і події;
* розширюваний набір елементів управління і бібліотек класів дозволяє швидше розробляти додатки;
* ASP.NET спирається на багатомовні можливості .NET, що дозволяє писати код сторінок на VB.NET, Delphi.NET, Visual C #, J # і т. д .;
* можливість кешування всієї сторінки або її частини для збільшення продуктивності;
* можливість кешування даних, що використовуються на сторінці;
* можливість поділу візуальної частини та бізнес-логіки у різні файли («code behind»);
* розширювана модель обробки запитів;
* розширена подієва модель;
* розширювана модель серверних елементів управління;
* наявність Master-сторінок для завдання шаблонів оформлення сторінок;
* підтримка CRUD-операцій при роботі з таблицями через GridView;
* вбудована підтримка AJAX.
  + 1. **СУБД MS SQL Server**

SQL Server є однією з найбільш популярних систем управління базами даних в світі. Дана СУБД підходить для самих різних проектів: від невеликих додатків до великих високонавантажених проектів. Для організації баз даних MS SQL Server використовує реляційну модель. Мова, що використовується для запитів – Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями [8]. Залежно від завдання, яку виконує команда T-SQL, вона може належати до одного з наступних типів:

* 1. DDL (Data Definition Language / Мова визначення даних). До цього типу належать різні команди, які створюють базу даних, таблиці, індекси, збережені процедури тощо.

1. DML (Data Manipulation Language / Мова маніпуляції даними). До цього типу відносять команди на вибір даних, їх оновлення, додавання, видалення – загалом всі ті команди, за допомогою яких ми можемо керувати даними.
2. DCL (Data Control Language / Мова управління доступу до даних). До цього типу відносять команди, які керують правами щодо доступу до даних.

MS SQL Server характеризується такими особливостями як:

1. Продуктивність. SQL Server працює дуже швидко.
2. Надійність і безпека. SQL Server надає шифрування даних.
3. Простота. З даної СУБД відносно легко працювати і вести адміністрування.

Одним з основних недоліків використання Microsoft SQL Server замість альтернативної системи управління реляційною базою даних є те, що варіанти ліцензування досить дорогі. Незважаючи на те, що використання програмного забезпечення для розробки чи освітніх цілей є безкоштовним, для використання будь-якого бізнесу використовується ліцензійна комісія. Для SQL Server 2008, наприклад, SQL Server Standard Edition коштує 7171$ доларів США за процесор. Видання Microsoft SQL Server Datacenter становить 54990$ доларів США за процесор, за даними веб-сайту Microsoft. Для малих підприємств та приватних осіб, які працюють на некомерційних веб-сайтах, це недоступне. Конкуруюче програмне забезпечення, таке як MySQL, часто є вільним для використання. Однак у випадках, коли це не так, найдорожчий пакет MySQL Enterprise коштує 4999$ на рік для кожного сервера. Це значно дешевше, ніж навіть стандартний пакет Microsoft SQL.

Microsoft SQL Server призначений тільки для роботи на серверах на базі Windows. З різних причин, включаючи витрати на ліцензування та проблеми безпеки, розробники можуть вибирати розміщення своїх веб-сайтів на машинах на базі Unix.У цьому випадку вони не зможуть використовувати SQL Server. Конкуруючі продукти часто можуть працювати на інших платформах. На відміну від Microsoft SQL Server, MySQL підтримується на всіх основних платформах, включаючи Windows, Linux, Mac OSX та інші варіанти Unix. На додаток до неможливості запускатись на платформах, відмінних від Windows, також можуть виникати проблеми сумісності з взаємодією з додатками, які працюють на інших платформах.

* + 1. **Microsoft Visual Studio**

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ [9].

Переваги Microsoft Visual Studio:

* + - 1. Вбудований Web-сервер. Для обслуговування веб-додатків ASP.NET необхідний Web-сервер, який очікує веб-запити і обробляє відповідні сторінки. Наявність в Visual Studio інтегрованого Web-сервера дозволяє запускати веб-сайт прямо з середовища проектування, а також підвищує безпеку, виключаючи можливість отримання доступу до тестового Web-сайту з будь-якого зовнішнього комп'ютера, оскільки тестовий сервер може приймати з'єднання тільки з локального комп'ютера.
      2. Підтримка безлічі мов при розробці. Visual Studio дозволяє писати код у своїй мові або будь-яких інших переважних мов, використовуючи весь час один і той же інтерфейс (IDE). Більше того, Visual Studio також дозволяє створювати веб-сторінки на різних мовах. Єдиним обмеженням є те, що в кожній Web-сторінці можна використовувати лише одну мову.
      3. Менше коду для написання. Для створення більшості додатків потрібна достатня кількість стандартного стереотипного коду, а також Web-сторінки ASP. NET тому не виключення. Наприклад, додавання Web-елемента управління, Приєднання обробників подій і корегування форматування вимагає встановлення в розмітці сторінки ряду деталей. У Visual Studio такі деталі встановлюються автоматично.
      4. Інтуїтивний стиль кодування. За замовчуванням Visual Studio форматує код за його введенням, автоматично вставляє необхідні відступи і застосовує колірне кодування для виділення елементів типу коментарів. Такі незначні відмінності роблять код більш зручним для читання та менш піддається помилці. Приймаються Visual Studio автоматично параметри форматування, які можна навіть налаштовувати, що дуже зручно.
      5. Більш висока швидкість розробки. Багато з функціональних можливостей Visual Studio направлені на те, щоб допомогти розробнику зробити свою роботу швидше. Зручні функції, такі як функції IntelliSense (яка здатна відловлювати помилки та пропонувати правильні варіанти), функції пошуку та заміни (яка дозволяє шукати ключові слова як в одному файлі, так і в цілому проекті) та функції автоматичного додавання та видалення коментарів (які можуть тимчасово закрити блоки коду), дозволяють розробнику працювати швидко і ефективно.
      6. Можливості налагодження. Параметри інструментів, що пропонуються в Visual Studio, є найкращим засобом для відстеження загадкових помилок та діагностики дивної поведінки.

Visual Studio також має й інші функції: можливість управління проектом; вбудована функція управління вихідним кодом; можливість рефакторизації коду; потужна модель розширюваності. Більше того, у випадку використання Visual Studio 2008 Team System розробник отримує розширені можливості для модульного тестування, спільної роботи та управління версіями коду (що значно більше того, що пропонується в простих інструментах, як Visual SourceSafe).

До недоліків Microsoft Visual Studio можна відзначити неможливість відладчика (Microsoft Visual Studio Debugger) відстеження в коді режиму ядра. Налагодження в Windows в режимі ядра в загальному випадку виконується при використанні WinDbg, KD або SoftICE.

* 1. **Опис архітектури веб-додатку**
     1. **Діаграма розміщення**

Діаграми розміщення (рис. 2.4) представляють фізичне розташування системи, показуючи, на якому фізичному обладнанні запускається та чи інша складова програмного забезпечення.

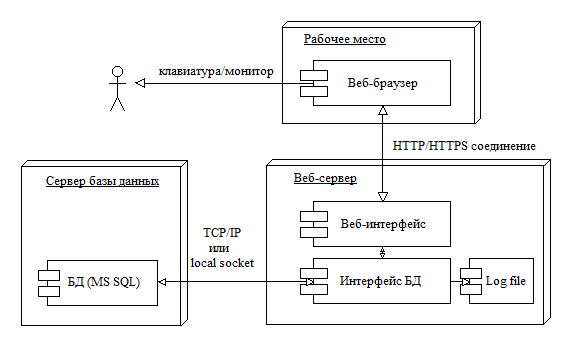


Рисунок 2.4 – Діаграма розміщення

Вузол (node) – це те, що може містити програмне забезпечення. Вузли бувають двох типів: пристрій і середовище виконання. Пристрій (device) – це фізичне обладнання: комп'ютер або пристрій, пов'язане з системою. Середовище виконання (execution environment) – це програмне забезпечення, яке може включати інше програмне забезпечення, наприклад операційну систему або процес-контейнер [10].

Вузли можуть містити артефакти (artifacts), які є фізичним уособленням програмного забезпечення; зазвичай це файли. Такими файлами можуть бути виконувані файли (такі як файли .eхе, виконавчі файли, файли DLL, файли JAR, складання або сценарії) або файли даних, конфігураційні файли, HTML-документи і т. д. Перелік артефактів всередині вузла вказує на те, що на даному вузлі артефакт розгортається в систему, яка запускається.

Артефакти можна зображувати у вигляді прямокутників класів або перераховувати їх імена всередині вузла.

Артефакти часто є реалізацією компонентів. Це можна показати, задавши значення-мітки всередині прямокутників артефактів.

Інформаційні шляхи між вузлами представляють обмін інформацією в системі. Можна супроводжувати ці шляхи інформацією про використання інформаційних протоколів.

* + 1. **Діаграма компонентів**

На діаграмі компонентів (рис. 2.5) показаний вид «білого ящика» внутрішньої структури двох пов'язаних підсистем – «Додавання нотатки» і «Об’єкти». В UML «підсистема» представляє собою стандартний компонентний стереотип для більших компонентів, які в свою чергу містять дрібніші компоненти.

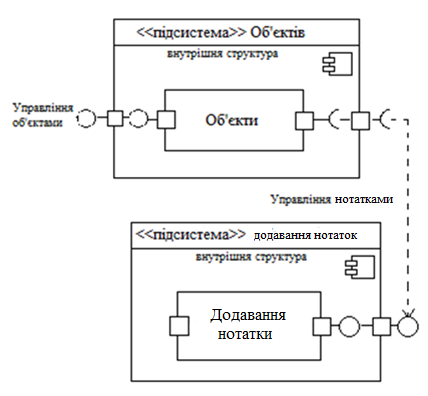


Рисунок 2.5 – Діаграма компонентів

Підсистема «Об’єкти» містить один компонент, пов'язаний з веб-додатком – «Об’єкти». Компонент «Об’єкти» дозволяє управляти елементами, використовуючи необхідний інтерфейс «Додавання нотатки», що надається компонентом «Додавання нотатки».

* 1. **Модель бази даних**

База даних, представлена у форматі Media Descriptor File (MDF), складається з однієї сутності: Notepad. Модель бази даних представлена на рисунку 2.6.

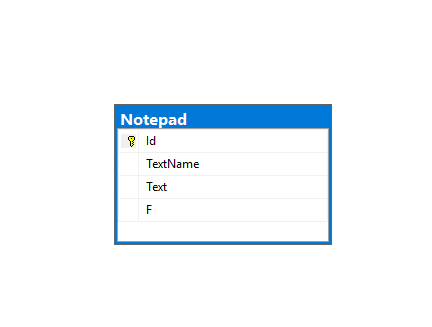


Рисунок 2.6 – Модель бази даних

Таблиця «Notepad» призначена для зберігання інформації про об’єкти веб-додатку. Кожний запис складається з наступних полів: Id, TextName, Text, F. Структура таблиці «Notepad» наведена в таблиці 2.1.

Таблиця 2.1 ­ – Опис структури таблиці «Notepad»

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Ім’я поля | Тип даних | Опис |
| PK | Id | Чисельний | ID нотатки |
|  | TextName | Текстовий | Ім’я нотатки |
|  | Text | Текстовий | Нотатка |
|  | F | Фото | Прикріплена фотографія |

Кожен рядок цієї таблиці ідентифікується особливим набором колонок який називається потенційним ключем. Поле Id і є цим ключем.

1. ОПИС ВЕБ-ДОДАТКУ
   1. **Тестування веб-додатку**
      1. **Верифікація та валідація веб-додатку**

При створенні нового об'єкта нерідко можуть виникати помилки. Але механізм валідації дозволяє інформувати користувача про помилки. Розглянемо html-сторінку з формою додання нотатки до системи (рис. 3.1–3.3).



Рисунок 3.1 – Реалізація форми додання нотатки до системи

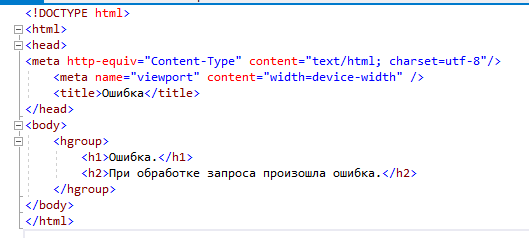


Рисунок 3.2 – Реалізація формування виключення

На сторінці (рис. 3.3) реалізується перевірка вірності введених даних нотатки, а саме:

* перевірка на підключення бази даних до додатку;
* перевірка на заповнення всіх полів форми.

У разі невиконання першої умови, система повідомляє спеціальним вікном (рис. 3.3).

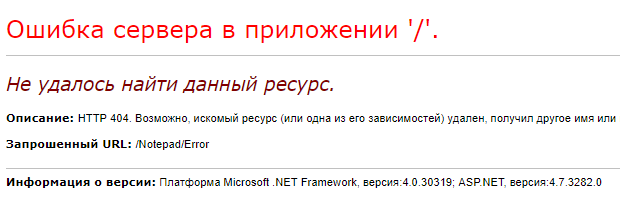


Рисунок 3.3 – Повідомлення про присутність об’єкта в системі

* + 1. **Ручне тестування**

Тестові випадки наведені в таблицях 3.1 – 3.8.

Таблиця 3.1 – Тестовий випадок №1

|  |  |
| --- | --- |
| Назва тесту | Тест на додання об’єкту з існуючим іменем в системі. |
| Описання | Тестування функцій додання об’єкту. |
| Тестові кроки | Введення існуючого об’єкту. |
| Очікуваний результат | Вдале додання об’єкту до системи. |
| Фактичний результат | Вдале додання об’єкту до системи. |
| Статус | Пройдений. |

Таблиця 3.2 – Тестовий випадок №2

|  |  |
| --- | --- |
| Назва тесту | Тест на додання об’єкту з неіснуючим іменем в системі. |
| Описання | Тестування функцій входу. |
| Тестові кроки | Тестування функцій додання об’єкту. |
| Очікуваний результат | Введення нового імені об’єкту. |
| Фактичний результат | Вдале додання об’єкту до системи. |
| Статус | Вдале додання об’єкту до системи. |

Таблиця 3.3 – Тестовий випадок №3

|  |  |
| --- | --- |
| Назва тесту | Тест на редагування імені об’єкту на існуюче ім’я. |
| Описання | Тестування редагування об’єкту в системі. |
| Тестові кроки | Введення існуючого імені в таблиці об’єктів. |
| Очікуваний результат | Вдале редагування імені. |
| Фактичний результат | Вдале редагування імені. |
| Статус | Пройдений. |

Таблиця 3.4 – Тестовий випадок №4

|  |  |
| --- | --- |
| Назва тесту | Тест на редагування імені об’єкту на неіснуюче ім’я. |
| Описання | Тестування редагування об’єкту в системі. |
| Тестові кроки | Введення неіснуючого імені в таблиці об’єктів. |
| Очікуваний результат | Вдале редагування імені. |
| Фактичний результат | Вдале редагування імені. |
| Статус | Пройдений. |

Таблиця 3.5 – Тестовий випадок №5

|  |  |
| --- | --- |
| Назва тесту | Тест на видалення об’єкту в системі. |
| Описання | Тестування видалення об’єкту в системі. |
| Тестові кроки | Видалення обраного об’єкту. |
| Очікуваний результат | Вдале видалення з системи. |
| Фактичний результат | Вдале видалення з системи. |
| Статус | Пройдений. |

Таблиця 3.6 – Тестовий випадок №6

|  |  |
| --- | --- |
| Назва тесту | Тест на додання нових нотаток. |
| Описання | Тестування функцій додання нотаток. |
| Тестові кроки | Створення нових нотаток. |
| Очікуваний результат | Вдале додання нотаток до системи. |
| Фактичний результат | Вдале додання нотаток до системи. |
| Статус | Пройдений. |

Таблиця 3.7 – Тестовий випадок №6

|  |  |
| --- | --- |
| Назва тесту | Тест на додання нових фотографії. |
| Описання | Тестування функцій додання фотографії. |
| Тестові кроки | Створення нової нотатки з фотографією. |
| Очікуваний результат | Вдале додання нотаток до системи. |
| Фактичний результат | Вдале додання нотаток до системи. |
| Статус | Пройдений. |

Перевіряючи працездатність всіх компонентів програми, ніяких дефектів в роботі програми не було виявлено.

* 1. **Керівництво користувача**
     1. **Загальний опис можливостей**

Для адміністрування веб-додатку, користувач має наступні можливості:

* додавати, видаляти, змінювати інформацію про нотатку;
* додавати та видаляти фотографію у збереженій нотатці;
* Перегляд збереженої інформації.
  + 1. **Об’єкти**

Користувач має право додання нових об’єктів до системи, їх редагування та видалення. Для виконання даних дій користувач має використати форму, яка зображена на рисунках 3.4–3.7.

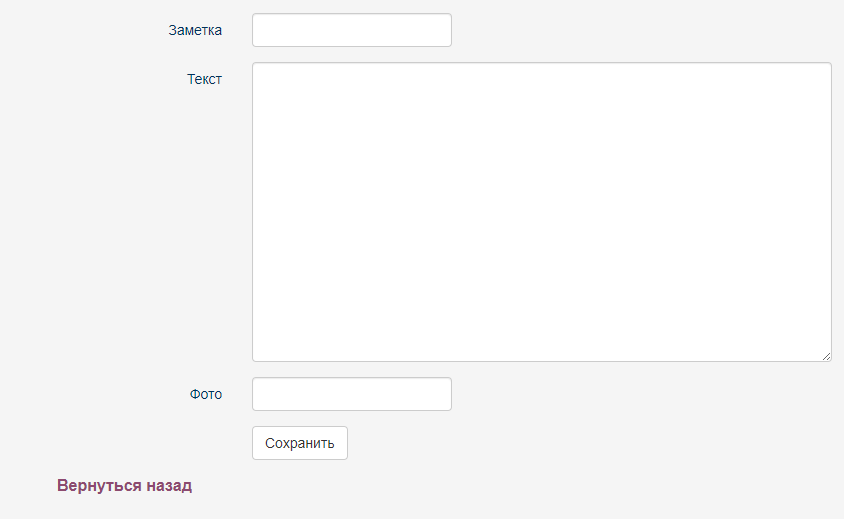


Рисунок 3.4 – Форма додавання нотатки до таблиці об’єктів

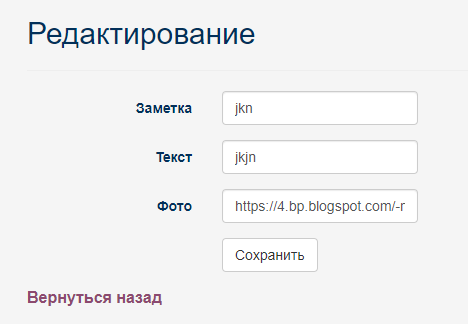


Рисунок 3.5 – Форма редагування нотатки в таблиці об’єктів

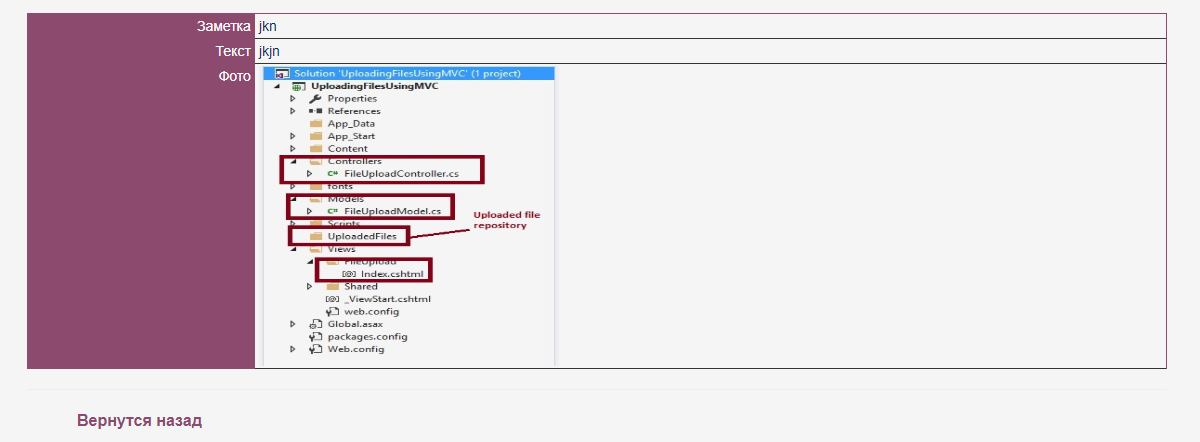


Рисунок 3.6 – Форма перегляду нотатки в таблиці об’єктів

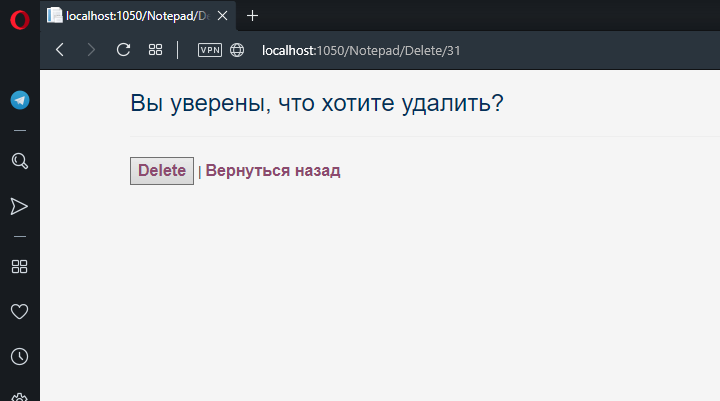


Рисунок 3.7 – Форма видалення нотатки з таблиці об’єктів

Було описано і протестовано можливості веб-додатку онлайн блокнота «Notepad». В описі продемонстровано такі можливості, як додавання, видалення, змінювання інформації про нотатку, всі ті функції, які потрібні для сучасного онлайн нотатника.

# ВИСНОВКИ

В ході виконання курсової роботи була досягнута основна мета роботи – розробка веб-додатку. Для досягнення поставленої мети були проведені вивчення і аналіз вже існуючої структури організації веб-додатків. При цьому було виявлено ряд існуючих недоліків існуючої систем, які можливо усунути за рахунок розробки та впровадження нового веб-додатку.

Онлайн блокнот «Notepad» призначений для швидкого створення та збереження нотатки з фотографією маючи лише доступ до Інтернету.

Розроблений веб-додаток є інформативним та лаконічним. Він є зручним для людей з різним ступенем комп’ютерної грамотності, маючи простий інтерфейс, який полегшує пошук інформації.

У розробленому веб-застосунку вдало поєднується інтерфейс додатку з його функціональністю і простотою використання. Блокнот дозволяє створювати, редагувати, переглянути та видалити нотатку.

Проаналізувавши досвід застосування інтернет-технологій для онлайн блокноту можна зробити висновок, що даний напрямок є дуже перспективним для ефективності вирішення проблеми швидкого створення та зберігання текстової інформації.

Результати перевірки показали, що система працює без помилок та є можливість роботи з веб-додатком у всіх популярних браузерах.

# СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. Онлайн блокнот «JustNotepad» [Електронний ресурс] – Режим доступу: URL – http://justnotepad.com/ru/, перегляд: 12.05.2019
2. Онлайн блокнот «Onlinenotepad.net» [Електронний ресурс] – Режим доступу: URL – https://onlinenotepad.net/ru/, перегляд: 12.05.2019
3. Онлайн блокнот «Pnote.ru» [Електронний ресурс] – Режим доступу: URL – http://pnote.ru/tree/index.php, перегляд: 12.05.2019
4. Онлайн блокнот «Notes.io» [Електронний ресурс] – Режим доступу: URL – http://itpride.net/useful/udobny-j-bloknot-onlajn.html, перегляд: 12.05.2019
5. HTML5 [Електронний ресурс] – Режим доступу: URL – https://ru.wikipedia.org/wiki/HTML5, перегляд: 13.05.2019
6. СSS [Електронний ресурс] – Режим доступу: URL – https://ru.wikipedia.org/wiki/CSS, перегляд: 13.05.2019
7. ADO.NET [Електронний ресурс] – Режим доступу: URL – https://ru.wikipedia.org/wiki/ADO.NET, перегляд: 13.05.2019
8. СУБД MS SQL Server [Електронний ресурс] – Режим доступу: URL – <https://www.microsoft.com/ru-ru/sql-server/sql-server-2016>, перегляд: 13.05.2019
9. Програма Visual Studio [Електронний ресурс] – Режим доступу: URL – <https://www.visualstudio.com/ru/>, перегляд: 25.04.2019
10. Вузол [Електронний ресурс] – Режим доступу: URL – https://uk.wikipedia.org/w/index.php?title=Вузол\_мережі&action=edit&redlink=1, перегляд: 25.04.2019