

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики

Кафедра теорії та технології програмування

Кваліфікаційна робота

на здобуття ступеня бакалавра

за спеціальністю 122 «Комп'ютерні науки»

на тему:

**РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ АВТОМАТИЗАЦІЇ
РОБОТИ ГУРТОЖИТКІВ**

Виконав студент 4-го курсу
Безух Дмитро Станіславович



Науковий керівник:
доцент, кандидат фіз.-мат. наук
Омельчук Людмила Леонідівна



(підпис)

Засвідчую, що в цій роботі немає запозичень
з праць інших авторів без відповідних
посилань.

Студент



Роботу розглянуто й допущено до захисту на
засіданні кафедри теорії та технології
програмування

« _____ » _____ 2021 р,

протокол № _____

Завідувач кафедри

Нікітченко М.С.

_____ (підпис)

Київ – 2021

РЕФЕРАТ

Обсяг роботи 44 сторінки, 23 ілюстрації, 8 таблиць, 16 джерел посилань.

БАЗА ДАНИХ, ІНТЕРФЕЙС ПРОГРАМНОГО ПРОДУКТУ, ТЕХНІЧНЕ ЗАВДАННЯ ДО ПРОДУКТУ, ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ, РОБОТА ГУРТОЖИТКУ.

Об'єктом даної роботи є процес організації роботи гуртожитків. Предметом роботи є вебзастосунок, для підтримки та надання актуальної інформації про функціонування гуртожитків закладу вищої освіти.

Метою роботи є проєктування та розробка вебзастосунку для автоматизації роботи гуртожитків Київського національного університету імені Тараса Шевченка: процесу поселення, переселення та виселення студентів із гуртожитків, організація роботи електронної черги на поселення в гуртожиток та ознайомлення студентів з інформацією про студентське містечко університету.

Методи розроблення: комп'ютерне моделювання, проєктування бази даних, розробка програмного продукту на основі моделі. Інструменти розроблення: інтегроване середовище розробки Visual Studio 2019, мова програмування C#, СКБД SQL Server.

Результати роботи: розглянуто процес організації роботи гуртожитків, проаналізовано наявні застосунки призначені для підтримки функціонування студентських містечок та гуртожитків, розроблено інтерфейс користувача для вебзастосунку, спроектовано та реалізовано програму для підтримки роботи гуртожитків Київського національного університету імені Тараса Шевченка.

Розроблений в дипломній роботі вебзастосунок може використовуватися студентами та адміністрацією Київського національного університету для організації роботи гуртожитків університету.

ЗМІСТ

СКРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	5
ВСТУП	6
РОЗДІЛ 1. ОГЛЯД НАЯВНИХ НА РИНКУ СИСТЕМ	8
1.1 Студентське містечко Київського національного університету імені Тараса Шевченка.....	8
1.2 Студентське містечко Київського політехнічний інституту ім. Ігоря Сікорського.....	9
1.3 Студентське містечко Тернопільського національного педагогічного університету імені Володимира Гнатюка.....	10
РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ.....	12
2.1 Технологія ASP.NET	12
2.2 Мова реалізації C#.....	14
2.3 Технологія AJAX.....	15
2.4 Технологія Razor.....	16
2.5 Фреймворк Bootstrap	17
РОЗДІЛ 3. ПРИЗНАЧЕННЯ ТА ЦІЛІ СТВОРЕННЯ ВЕБЗАСТОСУНКУ	18
4.1 Призначення вебзастосунку «Гуртожитки КНУТШ»	18
4.2 Цілі створення вебзастосунку «Гуртожитки КНУТШ»	18
4.3 Вимоги до вебзастосунку «Гуртожитки КНУТШ»	19
4.4 Технічні вимоги до вебзастосунку «Гуртожитки КНУТШ»	20
РОЗДІЛ 4. РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ	22
4.1 Реалізація бази даних	22
4.1.1 Логічна структура бази даних.....	22
4.1.2 Опис таблиць бази даних	23
4.2 Реалізація застосунку	26
РОЗДІЛ 5. ІНСТРУКЦІЯ КОРИСТУВАЧА	32
5.1 Інструкція неавторизованого користувача	32
5.2 Інструкція авторизованого користувача	36
5.3 Інструкція адміністратора	38
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	43

ДОДАТКИ	45
Додаток А. Діаграма бази даних	45
Додаток Б. Діаграма класів	46

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

ASP – Active Server Pages для .NET платформа розробки вебзастосунків;

CRUD – Create Read Update Delete, 4 основні функції управління даними;

MVC – Model-View-Controller, модель-представлення-контролер;

JIT – Just In Time, компіляція «на льоту»;

AJAX – Asynchronous JavaScript And XML, підхід до побудови користувацьких інтерфейсів вебзастосунків;

DHTML – динамічний HTML;

XML – Extensible Markup Language, мова розмітки;

DOM – Document Object Model, об'єктне представлення вмісту HTML-документу та інтерфейс для управління цим об'єктом;

CSS – Cascading Style Sheets, каскадні таблиці стилів;

Razor – синтаксис розмітки;

API – Application programming interface, прикладний програмний інтерфейс;

IDE – Integrated Design Environment, інтегроване середовище розробки;

C# – об'єктно-орієнтована мова програмування;

HTML – HyperText Markup Language, мова гіпертекстової розмітки;

C++ – мова програмування;

W3C – World Wide Web Consortium, головна міжнародна організація, що розробляє й впроваджує технологічні стандарти;

СКБД – система керування базами даних;

SQL – Structured Query Language, мова структурованих запитів.

ВСТУП

Оцінка сучасного стану об'єкта розробки. В наші дні мережа Інтернет набула широкого поширення. Завдяки цьому, з'явилася можливість швидкого і зручного доступу до різних джерел інформації. Інтернет має безліч переваг, таких, зокрема, як широкомасштабність і відносна доступність. Велика кількість людей використовують Інтернет для навчання, роботи і розваг, купівлі онлайн, а також для можливості переглядати перебіг необхідних процесів онлайн, наприклад електронних черг.

Більшість існуючих систем, що пов'язані зі студентським містечком університетів не автоматизують роботу гуртожитків, а лише надають детальну інформацію про них. Поселення займає велику кількість часу та створює величезні черги.

Актуальність роботи та підстави для її виконання. Нерідко трапляється ситуація, коли поселення/переселення в гуртожиток займає велику кількість часу та дуже часто не призводить до бажаного результату відразу, а потребує повторних витрат часу. Аналогічна ситуація складається із отриманням необхідної інформації про кімнату (кількість вільних місць, студентів, що в ній проживають тощо). Тому актуальним є створення універсального вебзастосунку, який буде автоматизувати роботу гуртожитків університету та надавати актуальну інформацію про них.

Мета й завдання роботи. Метою кваліфікаційної роботи є розробка програмного засобу для ознайомлення користувачів з інформацією про студентське містечко та автоматизації роботи гуртожитків. Для досягнення цієї мети поставлено такі завдання.

- Дослідити діяльність студентського містечка.
- Проаналізувати існуючі вебзастосунки, призначені для автоматизації роботи студентських містечок та визначити сильні та слабкі сторони кожного з них.

- Розробити вимоги до функціоналу продукту.
- Побудувати модель бази даних, яка відповідатиме всім вимогам, що висуваються до застосунку.
- Реалізувати вебзастосунок для автоматизації роботи гуртожитків Київського національного університету імені Тараса Шевченка.
- Поглибити та закріпити навички роботи з технологією MVC та взаємодією з базою даних.

Об'єкт, методи й засоби розроблення. Об'єктом розроблення вебзастосунку є організація процесу роботи гуртожитків Київського національного університету імені Тараса Шевченка.

Розробка вебзастосунку базувалася на кількох принципах. На початку розробки створюється початкова версія застосунку, яка оцінюється. На основі цієї оцінки допрацьовується додаток, з урахуванням всіх зауважень та побажань. Кожен етап розробки успадковує такий механізм, до того моменту поки не з'явиться додаток, який задовольняє всі поставлені вимоги.

В якості інструменту розробки програмного засобу було обрано Visual Studio 2019 – інтегроване середовище розробки (IDE) мовою програмування C#, яке є безкоштовним та вільно поширюваним.

РОЗДІЛ 1. ОГЛЯД НАЯВНИХ НА РИНКУ СИСТЕМ

Розглянемо вебзастосунки, які пов'язані з роботою гуртожитків закладів вищої освіти України, та проаналізуємо їх, щоб з'ясувати переваги та недоліки кожного та сформулювати вимоги до застосунку, що розробляється в рамках дипломної роботи.

1.1 Студентське містечко Київського національного університету імені Тараса Шевченка

На рис. 1 зображено вигляд сайту студентського містечка Київського національного університету імені Тараса Шевченка відразу після відкриття сайту [1]. Вгорі наявна панель для навігації сайтом. Вся необхідна інформація згрупована в розділи, цим самим спрощуючи пошук потрібної інформації.

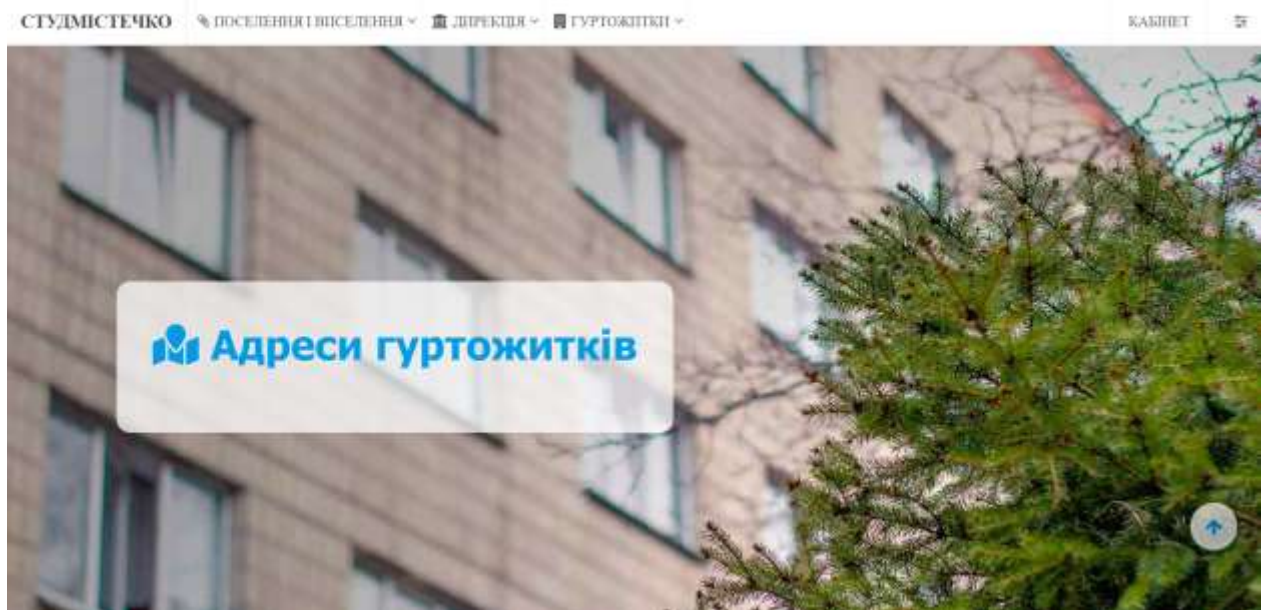


Рисунок 1 – Зовнішній вигляд сайту Студмістечко КНУ

На сайті студентського містечка наявна можливість входу в систему через кабінет користувача, в якому, після підтвердження проживання, з'являється інформація про період часу, на який активовано кабінет, можливість робити замовлення необхідних документів, послуг столяра чи електрика та перегляд історії замовлень (див. рис 2).

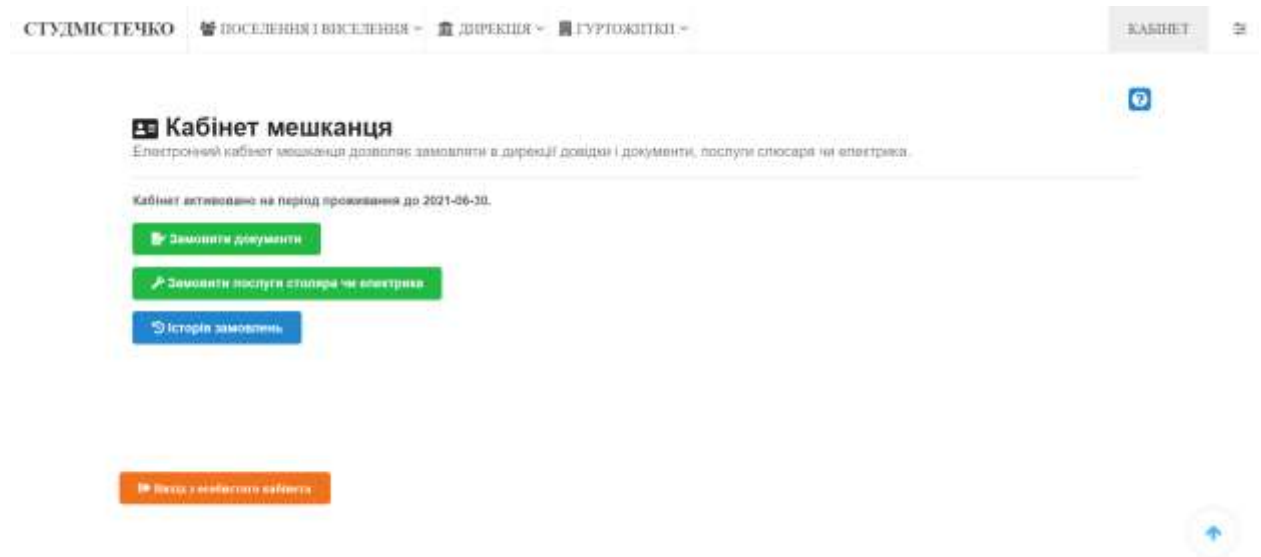


Рисунок 2 – Зовнішній вигляд кабінету користувача

Даний інтернет-сайт містить детальну інформацію про гуртожитки (опис, адреса, зображення), електронний кабінет, велику кількість інформації та зразків документів, проте відсутня можливість здійснювати необхідні дії (поселення, переселення, виселення) віддалено (електронно).

1.2 Студентське містечко Київського політехнічний інституту ім. Ігоря Сікорського

На рис. 3 зображено вигляд головної сторінки сайту студентського містечка Київського політехнічного інституту ім. Ігоря Сікорського [2]. Для зручності навігації по інтернет-сайту вся інформація згрупована в розділи, наявний опис гуртожитків, перелік документів та зразки бланків. Для

спрощення процедури оплати різних послуг, наявна можливість робити це онлайн.

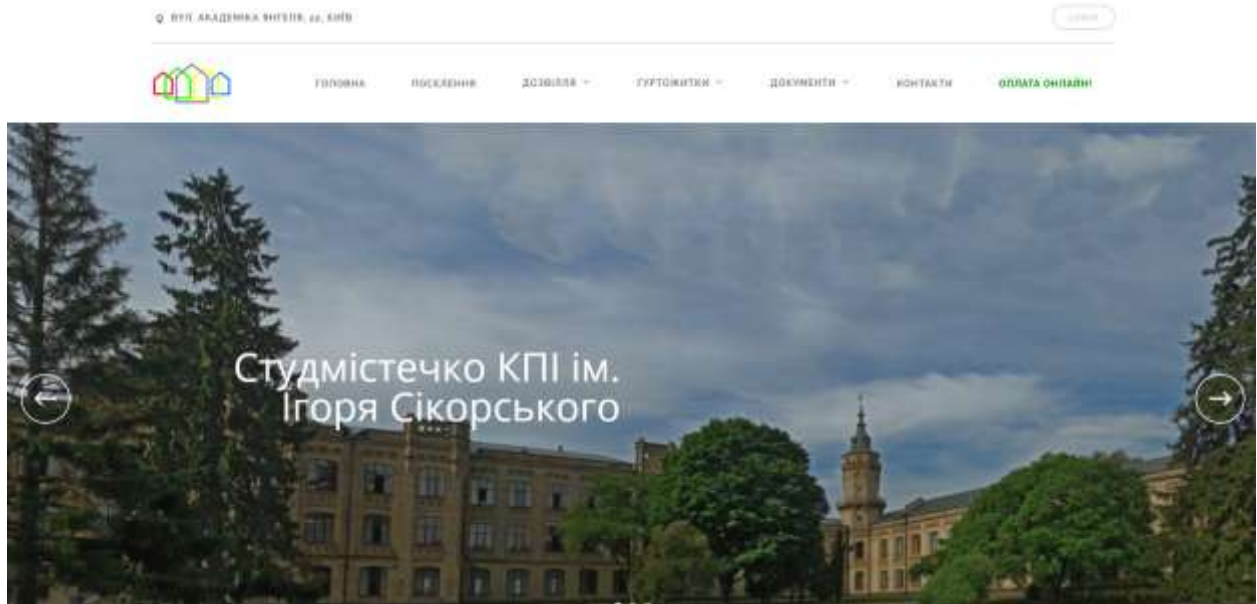


Рисунок 3 - Зовнішній вигляд сайту Студмістечко КПі ім. Ігоря Сікорського

1.3 Студентське містечко Тернопільського національного педагогічного університету імені Володимира Гнатюка

На рис. 4 зображено вигляд головної сторінки сайту студентського містечка Тернопільського національного педагогічного університету імені Володимира Гнатюка [3].

Вся інформація розташована на одній інтернет-сторінці, тобто не згруповано та не розділено на розділи, що ускладнює пошук необхідної інформації. Наявні посилання на накази про оплату проживання та коротка інформація про гуртожитки, включаючи невелику кількість зображень.



Рисунок 4 - Зовнішній вигляд сайту Студмістечко ТНПУ імені Володимира Гнатюка

Проаналізувавши існуючі інтернет-сайти, що надають інформацію про діяльність студентських містечок університетів, можна зробити висновок, що практично всі вони мають схожий функціонал, а саме надають певну інформацію про гуртожитки та зразки документів необхідних для поселення та виселення. Відрізняються роботою дизайнера та функціоналом електронних кабінетів, проте, жоден з них не надає можливість здійснювати електронне поселення чи переселення, відсутній доступ до черг та інформації про кімнати. Через відсутність таких можливостей, було прийнято рішення для реалізації власного застосунку.

РОЗДІЛ 2.

ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

Для розробки та реалізації вебзастосунку, для підтримки роботи гуртожитків, використано технології:

- технологія ASP.NET;
- мова реалізації C#;
- технологія AJAX;
- технологія Razor;
- фреймворк Bootstrap.

2.1 Технологія ASP.NET

Для створення вебзастосунку було обрано платформу ASP.NET. Microsoft ASP.NET Framework – це широко використовувана система розробки для реалізації вебдодатків високого рівня. Структура ASP.NET є популярною та відносно простою для розробників .Net, що призначена для створення програм ASP.NET за допомогою Visual Studio [4, 10].

Шаблон дизайну MVC (Model-View-Controller). Це загальний архітектурний шаблон, який використовується для проєктування та створення інтерфейсів та структури програми. Цей шаблон розділяє додаток на три частини, які є пов'язаними між собою. Ці конструкції використовуються для розрізнення подання даних від способу прийняття даних від користувача до їх відображення. Такий шаблон дизайну став поширеним у вебдодатках для розробки графічних інтерфейсів [5].

Сьогодні більшість програм дотримуються моделі MVC. Такий підхід дозволяє повторно використовувати код та здійснювати паралельну розробку, що робить роботу простішою та зрозумілішою. Компоненти,

створені за схемою проектування MVC, за своєю суттю не залежать один від одного. Ця функція допомагає розробникам легко та швидко використовувати компоненти та код одних додатків в інших.

Контролери (Controllers) – є одним з найважливіших компонентів програми MVC, оскільки він виконує роль координатора між представленням та моделлю, а також обробляє запити користувачів. Запит користувача отримує контролер, який вимагає від моделі виконання різних CRUD - операцій на основі запиту, а потім вирішує яке саме представлення необхідно повернути користувачеві. Насправді, вони контролюють загальний потік програми, беручи вхідні дані та надаючи належний результат.

Представлення (View) – це складова шаблону MVC, яка відповідає за інтерфейс користувача, відображає дані, що надходять від моделі власне інтерфейсу додатку, зазвичай, html-шаблон, який буде обов'язковим і відображатиме елементи керування html з даними.

Модель (Model) – центральним компонентом шаблону. Це динамічна структура даних програми, яка не залежить від інтерфейсу користувача, містить суто інформацію про логіку тих даних, з якими працює користувач. Модель не містить жодної інформації про те, як показати дані та залишається незмінною, незалежно від способу їх подання користувачеві [6].

Оскільки дані відстежуються представленням і контролюється спосіб їх подання користувачеві, розробник може використовувати подібні представлення з різними даними для різних додатків. Це економить багато часу та зусиль розробника.

В даний час існує велика кількість компаній, що розробляють програмне забезпечення, які обрали ASP.NET як основну технологію веброзробки.

Використання ASP.NET має наступні переваги:

1. ASP.NET – це технологія на стороні сервера, тому код обробляється на сервері Windows перед тим, як відобразитись у веб-браузері.

Тому програми ASP.NET виконуються швидше, ніж інтерпретовані сценарії.

2. Такі функції ASP.NET, як раннє прив'язування, компіляція JIT, послуги кешування та власна підтримка оптимізації, забезпечують високий рівень продуктивності.
3. Фреймворк ASP.NET не залежить від мови, тобто можна вибрати будь-яку мову програмування (C#, J#, VB тощо), яка найкраще підходить для вашої програми [10].
4. ASP.NET працює зі скомпільованим кодом. Завдяки чому отримує можливість ефективно використовувати різні механізми оптимізації коду.
5. Завдяки вбудованій інформації про конфігурацію ASP.NET легко розгорнути. Немає необхідності реєструвати компоненти, оскільки інформація про конфігурацію вбудована.
6. ASP.NET підтримується будь-яким браузером, запущеним на довільному пристрої.
7. Розробники можуть використовувати можливості, що надаються платформою .NET Framework при створенні додатків [9, 10, 12].

2.2 Мова реалізації C#

В якості мови реалізації вебзастосунку було обрано C#, враховуючи наступні переваги:

1. C# – це об'єктно-орієнтована мова, завдяки чому вона дозволяє створювати і підтримувані застосунки, модульні та багаторазові коди. Це одна з найважливіших переваг C# перед мовами C++.
2. C #, будучи мовою .NET, підтримує мовну сумісність, тобто C # може отримати доступ до коду, написаного будь-якою мовою, сумісною з .NET, а також може успадкувати класи, написані цими

мовами.

3. Мова C# дозволяє повною мірою використовувати всі можливості .Net Framework [6, 7].
4. Потужна бібліотека дає можливість C# будувати різні типи додатків, дозволяючи достатньо просто зберігати і отримувати інформацію з різних сховищ даних.
5. C# намагається спростити синтаксис, щоб бути більш послідовною і логічною, одночасно видаляючи деякі складні функції C++. Наприклад, C# позбавляється вказівників. Як типозахищена мова, C# не дозволяє безпосередньо маніпулювати пам'яттю, тому вказівники більше не потрібні в C# [4, 11].

2.3 Технологія AJAX

AJAX (асинхронний JavaScript та XML) – це метод створення інтерактивних програм, які негайно обробляють запити користувачів. AJAX поєднує кілька інструментів програмування, включаючи JavaScript, динамічний HTML (DHTML), розширювану мову розмітки (XML), каскадні таблиці стилів (CSS), об'єктну модель документа (DOM) [13].

AJAX дозволяє негайно оновлювати вміст на веб-сторінках, коли користувач виконує дію, на відміну від запиту HTTP, під час якого користувачі повинні чекати завантаження нової сторінки.

Додатки, створені за допомогою AJAX, використовують механізм, який діє як посередник між браузером користувача та сервером, з якого він запитує інформацію. Замість завантаження традиційної веб-сторінки браузер користувача використовує механізм AJAX, який довантажує необхідні користувачу дані. Механізм продовжує працювати у фоновому режимі, використовуючи JavaScript для зв'язку з веб-браузером. Введення користувачем даних або натискання на елементи сторінки надсилає виклик

JavaScript на механізм AJAX, який може реагувати миттєво у багатьох випадках. Якщо механізму потрібні додаткові дані, він запитує їх із сервера, як правило, використовуючи XML, одночасно оновлюючи сторінку [7].

Модель AJAX:

1. Користувач вводить деякі дані в поле для вводу та натискає якийсь елемент веб-сторінки.
2. Браузер надсилає запит на сервер.
3. Сервер повертає запитані користувачем дані.
4. Веб-сторінка відображує повернені від сервера дані, не оновлюючи сторінку.

Розробка сайту з використанням AJAX дає безліч переваг:

1. Зменшення навантаження на сервер.
2. Надає розробникам додатків незалежність від платформи.
3. Додаток на базі AJAX використовує меншу пропускну здатність сервера, оскільки немає необхідності перезавантажувати повну сторінку.

Для реалізації вебзастосунку було обрано технології, що найкраще підходять для реалізації поставлених задач, вирішено метод та протокол обміну інформацією між модулями та описано чому вибрані технології підходять під задачі проєкту. Також обрано архітектурні патерни [8, 13].

2.4 Технологія Razor

Razor – це синтаксис програмування для вбудовування серверного коду у код веб-сторінки.

Синтаксис Razor базується на платформі ASP.NET, яка спеціально розроблена для створення веб-додатків. Веб-сторінки Razor описуються як HTML-сторінки з двома видами вмісту: вмістом HTML та кодом Razor.

Перед відправленням сторінки HTML у браузер, сервер спочатку її читає, знаходить код Razor та виконує його.

Код, який виконується на сервері, може виконувати завдання, які неможливо виконати в браузері, наприклад, доступ до бази даних сервера. Код сервера може створювати динамічний вміст HTML на льоту, перед тим, як його відправити в браузер. З погляду браузера, сформований серверним кодом HTML, нічим не відрізняється від статичного вмісту HTML [14].

2.5 Фреймворк Bootstrap

Bootstrap – це потужний набір інструментів з відкритим кодом – інтерфейсна бібліотека HTML, CSS та JavaScript для створення вебсторінок та вебдодатків.

Після випуску з відкритим кодом у 2011 році Bootstrap став дуже популярним і не без підстав. Вебдизайнери та веброзробники використовують Bootstrap, оскільки він гнучкий і простий у роботі. Основні його переваги в тому, що він адаптивний за дизайном, підтримує широку сумісність з браузерами, пропонує послідовний дизайн за допомогою повторно використовуваних компонентів, простий у використанні та швидкий у навчанні. Він пропонує широку розширюваність за допомогою JavaScript, оснащений вбудованою підтримкою плагінів jQuery та програмним API JavaScript. Bootstrap можна використовувати з будь-якою IDE або редактором, а також будь-якою технологією та мовою на стороні сервера, починаючи від ASP.NET і закінчуючи Ruby on Rails [15].

РОЗДІЛ 3.

ПРИЗНАЧЕННЯ ТА ЦІЛІ СТВОРЕННЯ ВЕБЗАСТОСУНКУ

4.1 Призначення вебзастосунку «Гуртожитки КНУТШ»

Призначенням вебзастосунку «Гуртожитки КНУТШ» є використання проєкту для детального ознайомлення з інформацією про гуртожитки, автоматизації процесу поселення, переселення студентів, надання інформації про кімнати та можливості переглядати електронні черги на поселення.

Кваліфікаційна бакалаврська робота передбачає:

- аналіз методів, методик, технологій і моделей, що використовуються для створення вебзастосунів;
- дослідження діяльності студентського містечка Київського національного університету імені Тараса Шевченка;
- аналіз існуючих програмних засобів;
- проєктування та програмну реалізацію застосунку.

4.2 Цілі створення вебзастосунку «Гуртожитки КНУТШ»

Вебзастосунок «Гуртожитки КНУТШ» створюється з метою:

- ознайомлення студентів зі студентським містечком Київського національного університету імені Тараса Шевченка;
- автоматизація поселення, переселення, виселення;
- надання актуальної інформації про стан гуртожитків та можливість зворотнього зв'язку;
- забезпечення зручного керування даними застосунку.

4.3 Вимоги до вебзастосунку «Гуртожитки КНУТШ»

Застосунок повинен бути розроблений так, щоб користувачу було інтуїтивно зрозуміло як користуватися сайтом та мати приємний дизайн.

Можна виділити три типи користувачів: звичайні відвідувачі, студенти та адміністратори.

Звичайні відвідувачі, переглядають малу кількість сторінок і залишаються на них протягом короткого проміжку часу. Такі відвідувачі найменше зацікавлені в інформаційному наповненні сайту. Але їх можна зацікавити зображеннями, дизайном, різними елементами управління і персоналізованим підходом.

Студенти запитують більше число сторінок і залишаються на сайті протягом більшої кількості часу, ніж відвідувачі. Мають менший інтерес до зображень і дизайну, їх більше цікавить інформативність сайту та його можливості.

Адміністратори мають доступ до всієї інформації на сайті, контролюють її коректність та оновлюють за потреби. Мають можливість редагувати базу даних, та здійснювати підтвердження користувачів, поселення/переселення/виселення.

Вебзастосунку передбачає виділення наступних функціональних підсистеми:

- адміністративна, призначена для керування даними на сайті, відповіді на повідомлення, підтвердження користувачів та запитів на поселення, переселення та виселення;
- підсистема студента, призначена для перегляду вмісту сайту, зворотнього зв'язку з адміністрацією, бронювання місця в черзі на поселення/переселення, контролю правильності виконання поселення та перегляд історії перебування в гуртожитках;
- підсистема звичайного відвідувача, призначена для перегляду

інформації про студентське містечко Київського національного університету імені Тараса Шевченка.

4.4 Технічні вимоги до вебзастосунку «Гуртожитки КНУТШ»

В даний час все більше і більше людей користуються Інтернетом - це призвело до того, що стало з'являтися багато різних браузерів (основний інструмент користувача Інтернетом). Поява великої кількості браузерів призвело до проблеми сумісності. Хоча, кожен з розробників браузерів і вважає себе найкращим, але, тим не менш, єдиного стандарту для всіх немає.

Часто можна зустріти сайт, вид сторінок якого буде кардинально відрізнятися в різних браузерах. Ці проблеми призвели до виникнення стандарту W3C. У цьому стандарті містяться тільки ті інструменти, які підтримуються всіма браузерами (більшістю). Грунтуючись на статистиці (на момент написання курсової), більшість користувачів Інтернету використовують такі браузери:

Google Chrome	63.58 %;
Safari	19.19%;
Firefox	3.82 %;
Samsung Internet	6.13 %;
Edge	2.97 %;
Opera	2.11 %.

На рис. 5 зображено статистику використання браузерів [16].

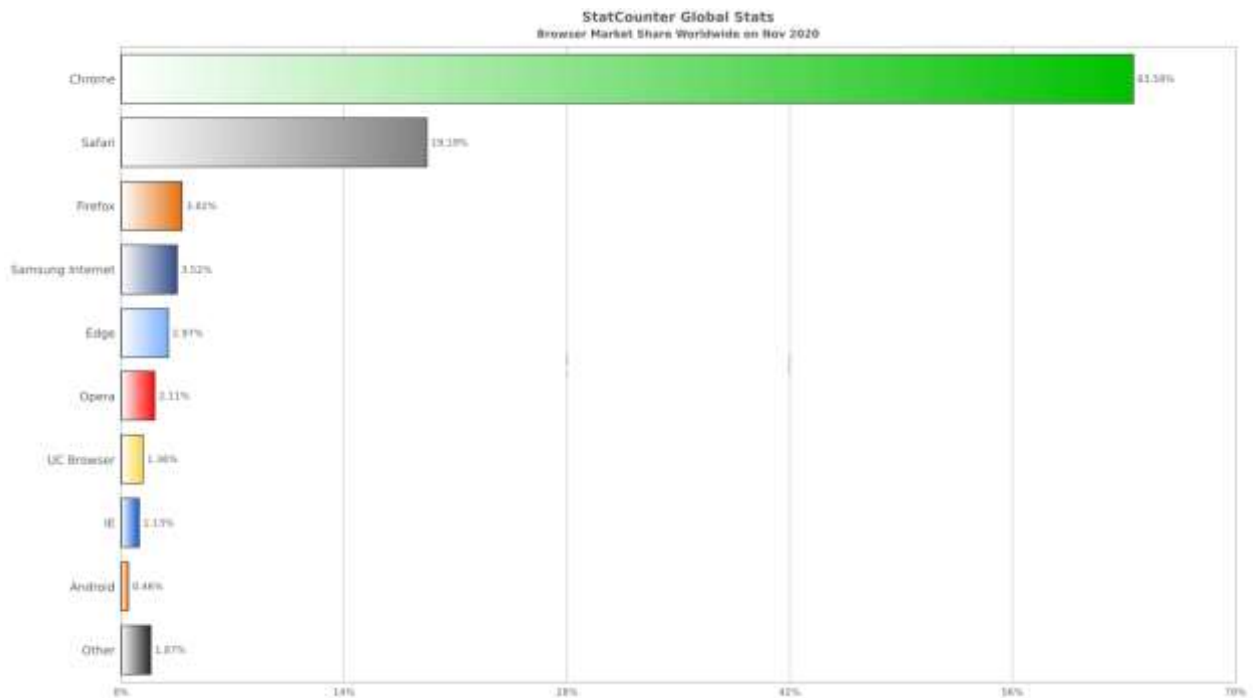


Рисунок 5 – Статистика використання браузерів

Розробка проєкту повинна орієнтуватися на браузери Google Chrome, Firefox, Edge і Opera. Таким чином, щоб в цих браузерах дизайн відображався однаково, і вся функціональність працювала коректно.

Застосунок повинен бути розроблений використовуючи технологію ASP.NET. Джерелом даних повинна бути СКБД, як сервер бази даних використовувати Microsoft SQL Server.

На довільні помилкові дії користувача, пов'язані з введенням некоректних даних, не заповненням (пропуском) обов'язкових полів введення в формах та інші, які можуть бути оброблені застосунком, генеруються відповідні повідомлення про помилки українською мовою, в межах загального дизайну сайту.

РОЗДІЛ 4. РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

4.1 Реалізація бази даних

4.1.1 Логічна структура бази даних

База даних розроблюваного інтернет-сайту складається з 8 таблиць, структура яких зображена на діаграмі бази даних (див. додаток А.1). Опис кожної з таблиць наведено у табл. 1.

Таблиця 1 – Короткий опис таблиць бази даних

Номер	Таблиця	Опис
1	Hostels	Таблиця для збереження інформації про гуртожитки.
2	Photo_for_hostel	Таблиця для збереження фотографій гуртожитку.
3	Appeals	Таблиця для збереження скарг, пропозицій від користувачів.
4	Histories	Таблиця для збереження історії проживання користувача.
5	Users	Таблиця для збереження інформації про користувачів застосунку.
6	Roles	Таблиця ролей застосунку.
7	Rooms	Таблиця для збереження інформації про кімнати гуртожитку.
8	Student_room	Таблиця для збереження відповідності між користувачем та кімнатою.

4.1.2 Опис таблиць бази даних

Таблиця `dbo.Hostels` (див. табл. 2) містить унікальний номер гуртожитку, коротку інформацію, адресу, основне зображення гуртожитку та координати для відображення місцезнаходження на карті. Додаткові зображення та їх зв'язок із гуртожитком зберігається в табл. 3. Кожен гуртожиток містить кімнати, інформація про які зберігається в табл. 4.

Таблиця 2 – Структура таблиці `dbo.Hostels`

Атрибут	Тип	Опис
ID	int	Ідентифікатор запису в таблиці.
Name	nvarchar(MAX)	Назва гуртожитку.
Address	nvarchar(MAX)	Адреса гуртожитку.
Latitude	float	Координата знаходження гуртожитку – широта.
Longitude	float	Координата знаходження гуртожитку – довгота.
Number_of_floors	int	Кількість поверхів.
Description	nvarchar(MAX)	Опис.
Image	varbinary(MAX)	Зображення гуртожитку.

Таблиця 3 – Структура таблиці `dbo.Photo_for_hostel`

Атрибут	Тип	Опис
ID	int	Ідентифікатор запису в таблиці.
Image	varbinary(MAX)	Зображення.
HostelID	int	Ідентифікатор гуртожитку.

Таблиця 4 – Структура таблиці dbo.Rooms

Атрибут	Тип	Опис
ID	int	Ідентифікатор запису в таблиці.
Number	int	Номер кімнати.
Number_of_seats	int	Загальна кількість місць.
Filled_seats	int	Кількість зайнятих місць.
Sex	int	Стать, користувачі якої можуть проживати в кімнаті.
HostelID	int	Ідентифікатор гуртожитку.

Таблиця dbo.Student_room (див. табл. 5) зберігає інформацію про зв'язок студента із гуртожитком (кімнатою). Також, для кожного користувача зберігаються відповідні записи (історія) при поселенні, переселенні або виселенні (див. табл. 6).

Таблиця 5 – Структура dbo.Student_room

Атрибут	Тип	Опис
ID	int	Ідентифікатор запису в таблиці.
Registration_date	datetime	Дата поселення в кімнату.
Note	nvarchar(MAX)	Нотатки для запису особливої інформації проживання користувача.
IsConfirmed	bool	Підтвердження користувача.
UserID	int	Ідентифікатор користувача.
RoomID	int	Ідентифікатор кімнати.

Таблиця 6 – Структура таблиці dbo.Histories

Атрибут	Тип	Опис
ID	int	Ідентифікатор запису в таблиці.

Registration_date	datetime	Дата поселення в кімнату, переселення або виселення.
Note	nvarchar(MAX)	Нотатки для запису особливої інформації проживання користувача.
Status	int	Поле для деталізації діяльності в гуртожитку. Містить одне з трьох значень: поселення, переселення або виселення.
UserID	int	Ідентифікатор користувача.
RoomID	int	Ідентифікатор кімнати.

Таблиця `dbo.Users` (див. табл. 7) містить інформацію для входу користувачів до застосунку. Адміністратори мають доступ до керування даними на сайті, авторизовані користувачі можуть поселятися/переселятися до кімнати. Ролі, наявні в застосунку, зберігаються в таблиці `dbo.Roles` (див. табл. 8). В кожного авторизованого користувача є можливість залишати скарги, побажання, рекомендації, які зберігаються в таблиці `dbo.Appeals` (див. табл. 9).

Таблиця 7 – Структура `dbo.Users`

Атрибут	Тип	Опис
ID	int	Ідентифікатор запису в таблиці.
Name	nvarchar(MAX)	Ім'я користувача.
SurName	nvarchar(MAX)	Прізвище користувача.
Email	nvarchar(MAX)	Електронна пошта.
Password	nvarchar(MAX)	Пароль.
Date_end_studyin g	datetime	Дата закінчення навчання.

Sex	int	Стать.
IsConfirmed	bool	Підтвердження користувача.
RoleId	int	Ідентифікатор ролі.

Таблиця 8 – Структура dbo.Roles

Атрибут	Тип	Опис
ID	int	Ідентифікатор запису в таблиці.
Name	nvarchar(MAX)	Назва ролі користувача в застосунку.

Таблиця 9 – Структура таблиці dbo.Appeals

Атрибут	Тип	Опис
ID	int	Ідентифікатор запису в таблиці.
Title	nvarchar(MAX)	Заголовок звернення.
Description	nvarchar(MAX)	Основний текст звернення.
UserID	int	Ідентифікатор користувача.

4.2 Реалізація застосунку

Модель проєкту складається з 10 класів та двох допоміжних множин (див. додаток Б.1) для фіксування можливих значень статі та статусу поселення. Опис та призначення кожного класу наведений в табл. 7.

Таблиця 7 – Опис класів моделей

Номер	Клас	Опис
1	Appeal	Модель даних для збереження скарг, побажань, питань та зворотнього зв'язку між користувачами застосунку та

		адміністраторами.
2	History	Модель даних для фіксування історії перебування користувачів в гуртожитках.
3	Hostel	Описує сутність «гуртожиток», містить інформації про кімнати та є однією з основних сутностей застосунку.
4	LoginModel	Валідована модель для входження користувача до застосунку.
5	RegisterModel	Валідована модель для реєстрації користувачів. Основна відмінність від моделі User – це наявність додаткових полів для повторного введення електронної пошти та паролю.
6	Photo_for_hostel	Модель даних для збереження додаткових зображень гуртожитків.
7	Room	Модель для опису інформації про кімнату, включаючи кількість зайнятих місць.
8	Student_room	Модель, що описує зв'язок користувачів застосунку з кімнатами.
9	User	Модель даних для збереження інформації про користувачів.
10	Role	Модель, призначена для зберігання можливих ролей застосунку.

У табл. 8 наведено основні методи управління даними застосунку та описано функції, які вони виконують.

Таблиця 8 – Опис основних методів застосунку

Номер	Клас	Метод	Опис
-------	------	-------	------

1	AccountController	Login	Виконує пошук зареєстрованого користувача застосунку за введеними даними та, у разі успіху, авторизує його.
2	AccountController	Register	Перевіряє збіг введеної електронної пошти з наявними в застосунку та, у випадку коли збіг відсутній – реєструє користувача.
3	AccountController	Edit_Admin	Виконує підтвердження зареєстрованих користувачів на роль адміністратора.
4	AccountController	Index	Знаходить всіх користувачів застосунку та передає знайдені дані до представлення.
5	AppealsController	Create_appeal	Записує в базу даних повідомлення від користувачів.
6	AppealsController	Get_Appeals_for_User	Знаходить та повертає представленню список звернень користувача за його електронною адресою.
7	AppealsController	Write_Answer	Оновлює запис в таблиці звернень, додаючи вказану

			відповідь та надсилає її на пошту користувача.
8	HistoriesController	Users_history	Знаходить історію проживання користувача та повертає результат представленню.
9	Photo_for_hostelController	Create	Завантажує зображення та прив'язує його до гуртожитку.
10	Photo_for_hostelController	Index_for_Hostel	За ідентифікатором гуртожитку знаходить додаткові фотографії та повертає їх частковому представленню.
11	RoomsController	Rooms_for_Hostel	За ідентифікатором гуртожитку знаходить кімнати, та визначає можливість поселення, переселення в них авторизованого користувача.
12	RoomsController	Create_rooms_for_hostel	Створює кімнати для заданого гуртожитку у введеному адміністратором діапазоні.
13	RoomsController	Users_for_room	За ідентифікатором кімнати знаходить користувачів, які стоять в черзі на поселення та

			передає знайдений список частковому представленню.
14	RoomsController	Users_who_live	Знаходить користувачів, що проживають в необхідній кімнаті та повертає результат представленню.
15	RoomsController	Details	Знаходить необхідну інформацію про кімнату та визначає можливість поселення/переселення та відношення авторизованого користувача щодо неї.
16	Student_roomController	Edit	Підтверджує поселення до кімнати та робить необхідний запис в історію (поселення або переселення).
17	Student_roomController	Create_StR	Записує користувача в чергу на поселення або переселення до кімнати.
18	Student_roomController	DeleteConfirmed	Видаляє зв'язок між користувачем та кімнатою і робить відповідний запис в історію про виселення.
19	HomeController	Info	Представлення цього методу відображає

			інструкцію поселення та виселення з кімнати.
20	HomeController	SendMessage	Статичний метод, що реалізовує відправлення повідомлення на пошту.

РОЗДІЛ 5. ІНСТРУКЦІЯ КОРИСТУВАЧА

5.1 Інструкція неавторизованого користувача

Коли користувач вперше заходить на сайт, перед ним відображається головна сторінка (див. рис. 6), на якій можна побачити коротку інформацію та статистику про студентське містечко. Вся інформація згрупована по розділах, посилання на які знаходяться на панелі вгорі. Оскільки користувач не авторизований, то є можливість увійти до застосунку або зареєструватися (див. рис. 7).

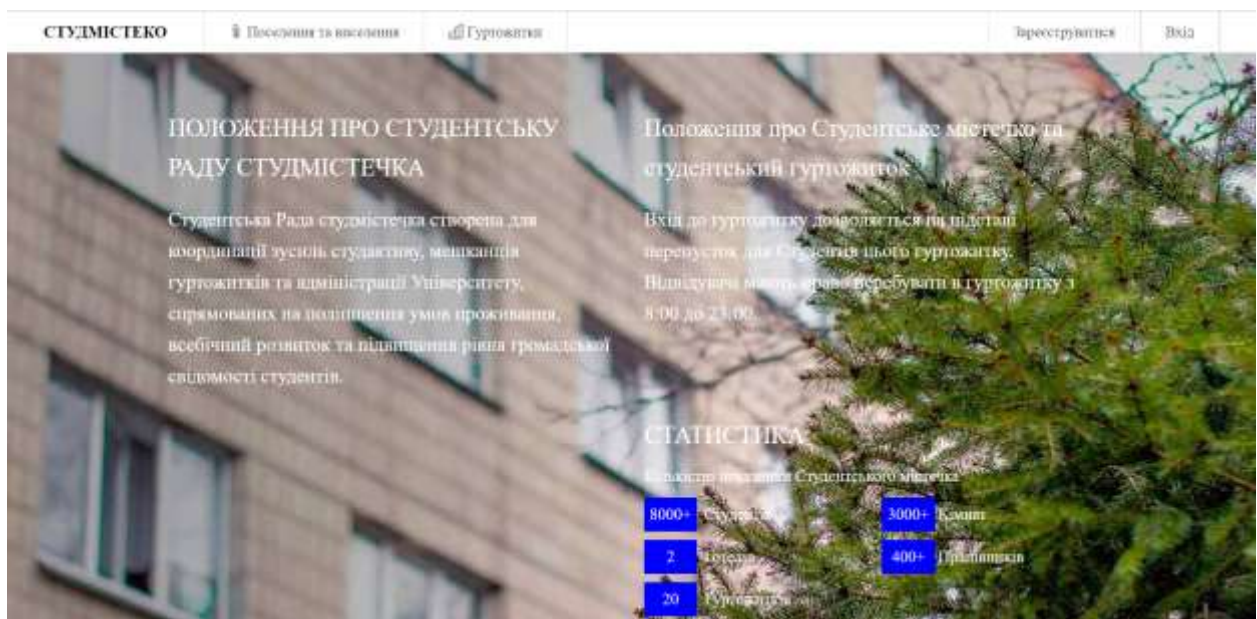


Рисунок 6 – Головна сторінка

Після реєстрації, адміністратор повинен підтвердити акаунт після чого користувач зможе увійти до застосунку. Якщо акаунт не буде підтверджений, то при спробі входу, виведеться відповідне повідомлення (див. рис. 8).

СТУДМІСТЕКО	Поселення та виселення	Гуртожитки	Зареєструватися	Вхід
-------------	------------------------	------------	-----------------	------

Реєстрація користувача:

Ім'я	Прізвище
<input type="text"/>	<input type="text"/>
Адреса електронної пошти	Підтвердити електронну адресу
<input type="text"/>	<input type="text"/>
Пароль	Підтвердити пароль
<input type="text"/>	<input type="text"/>
Дата закінчення навчання	Стать
<input type="text" value="ДД.ММ.ПП"/>	<input type="text" value="М"/>

[ЗАРЕЄСТРУВАТИСЯ](#)

Рисунок 7 – Сторінка реєстрації

СТУДМІСТЕКО	Поселення та виселення	Гуртожитки	Зареєструватися	Вхід
-------------	------------------------	------------	-----------------	------

Вхід в електронний кабінет мешканця студмістечка

• Користувач не підтверджений

Електронна адреса
<input type="text" value="olga_bezukh@knu.ua"/>
Пароль
<input type="text" value="*****"/>

[УВІЙТИ](#)

Рисунок 8 – Повідомлення при спробі увійти до застосунку, використовуючи не підтверджений акаунту

Перейшовши до розділу «Поселення та виселення», відкриється сторінка, на якій знаходиться детальна інструкція для поселення до кімнати та виселення з неї, включаючи зразки необхідних заяв та приклади їх заповнення (див. рис. 9).

СТУДМІСТЕКО | Поселення та виселення | Гуртожитки | Зареєструватися | Вхід

Поселення до гуртожитку

Алгоритм поселення до студентських гуртожитків Київського національного університету імені Тараса Шевченка на 2020/2021 навчальний рік:

1. Крок 1. Удома:

- Якщо Ви рекомендовані до зарахування на навчання та надаете оригінали документів, і потребуєте поселення до гуртожитку, підготуйте та зберігайте та візьміть із собою пакет документів, необхідних для поселення:
 1. Заява на поселення (роздрукувати ДОДАТОК №1 та заповнити за зразком ДОДАТОК №2).
 2. Копії документів, які підтверджують пільги (за їх наявності).
 3. Копія паспорту (паспорт зразка книжечка 1, 2 та 12 сторінок, паспорт зразка ID картка 1, 2 сторони, копія довідки з місця реєстрації (форма-131 – лише для паспорту ID).
 4. Копія реєстраційного номеру облікової картки платника податків (раніше індивідуальний податковий номер, ідентифікаційний код).
 5. Оригінал довідки від сімейного лікаря про проходження флюорографічного огляду, який проведено не більше року тому.
 6. Фотокартки 3 x 4 — 6 шт.
- Копії та оригінали документів, які підтверджують пільги, необхідно надати **ОБОВ'ЯЗКОВО!**

Наголосимо, німається з реєстрації місця проживання та військового обліку (для чоловіків) НЕ ОБОВ'ЯЗКОВО! Проживання в гуртожитку можливе без реєстрації місця проживання за адресою гуртожитку.

2. Крок Комісії з поселення:

- Комісії з поселення факультету/інституту збираються на кожному факультеті/інституті.
- Якщо Ви потребуєте поселення, то надаете комісії з поселення такі документи:
 1. Заява на поселення, заповнена за зразком (єдина встановлена форма).

Рисунок 9 – Алгоритм дій при поселенні та виселенні

Перейшовши до розділу «Гуртожитки», відкриється сторінка, на якій знаходиться перелік всіх гуртожитків студентського містечка (див. рис. 10), частина інформації про них та можливість перейти на сторінку вибраного гуртожитку.

СТУДМІСТЕКО | Поселення та виселення | Гуртожитки | Зареєструватися | Вхід

Гуртожитки Київського національного університету імені Тараса Шевченка




		
<p>Гуртожиток №1 вул. Ломоносова, 35</p> <p>В гуртожитку проживають студенти факультету радіофізики, електроніки та інформаційних систем, а також студенти юридичного факультету (5 поверх).</p>	<p>Гуртожиток №2 вул. Ломоносова 37</p> <p>В гуртожитку проживають студенти філософського факультету та факультету психології.</p>	<p>Гуртожиток №3 вул. Ломоносова, 41</p> <p>В гуртожитку проживають курсанти військового інституту, факультету спеціальності.</p>

Рисунок 10 – Перелік гуртожитків студентського містечка

На сторінці гуртожитку знаходиться детальна інформація про нього, розташування на карті та додаткові фотографії (див. рис. 11-12).

СТУДМІСТЕКО | Поселення та виселення | Гуртожитки | Зареєструватися | Вхід

Гуртожиток №1



В гуртожитку проживають студенти факультету радіофизик, електроніки та комп'ютерних систем, а також студенти юридичного факультету (5 поверх).


Кількість поверхів: 5
Кількість місць: 18
Адреса: вул. Ломоносова, 35

Розташування гуртожитку



Рисунок 11 – Детальна інформація про гуртожиток

СТУДМІСТЕКО | Поселення та виселення | Гуртожитки | Зареєструватися | Вхід



Повернутися

Рисунок 12 – Додаткові зображення гуртожитку

5.2 Інструкція авторизованого користувача

Після авторизації користувач може увійти до електронного кабінету (див. рис. 13), та з'являється можливість написати звернення (повідомлення) до адміністрації (див. рис. 14). В кабінеті користувача знаходиться його детальна інформація, історія перебування в гуртожитках та можливість завантажити всі звернення до адміністрації.

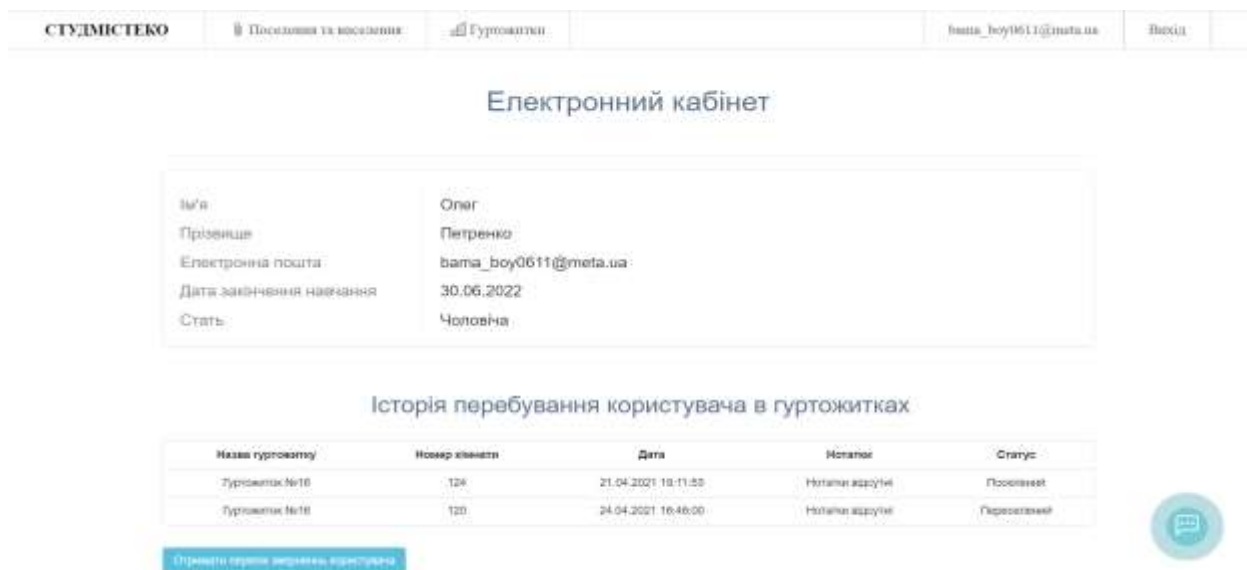


Рисунок 13 – Електронний кабінет

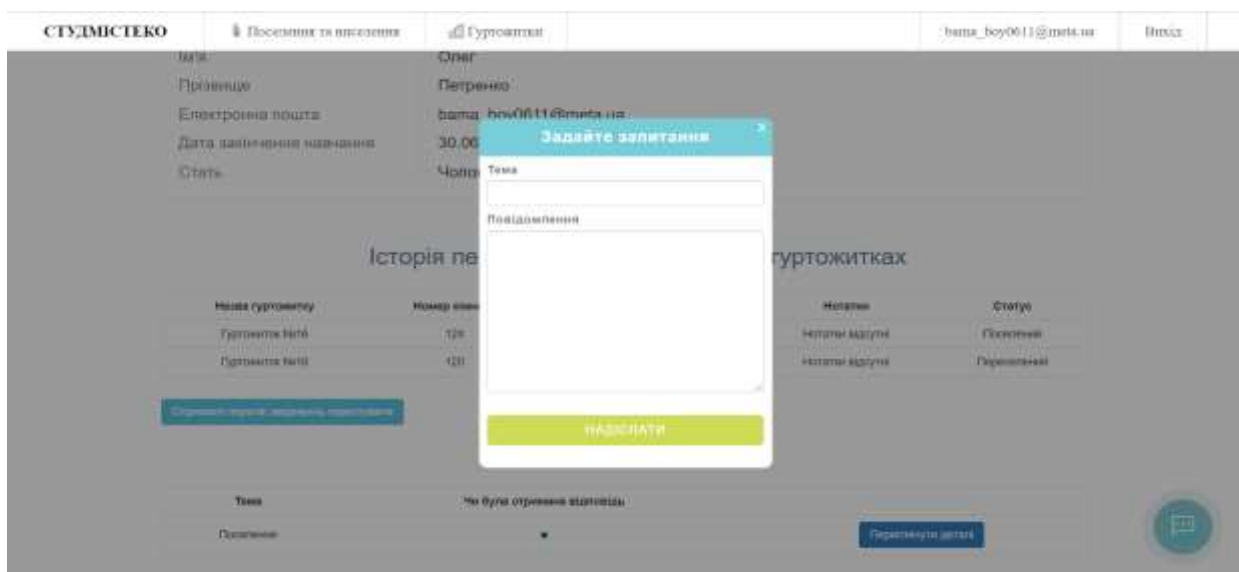


Рисунок 14 – Форма для зворотнього зв'язку

Після авторизації, на сторінці гуртожитку відображається список кімнат, коротка інформація та статус користувача відносно кожної з кімнат (див. рис. 15).

Кімнати

Номер кімнати	Стать, користувачі якої можуть поселитися	Максимальна кількість місць	Статус
100	Чоловіча	3	Проживає
101	Чоловіча	3	Наразі вільні місця
102	Чоловіча	3	Наразі вільні місця
103	Чоловіча	3	Наразі вільні місця
104	Чоловіча	3	Наразі вільні місця
105	Чоловіча	3	Наразі вільні місця

Поворотні кнопки: Повернутись, повідомлення.

Рисунок 15 – Перелік кімнат гуртожитку

Перейшовши на сторінку кімнати, можна переглянути детальну інформацію, студентів, які проживають в кімнаті та можливість слідкувати за чергою на поселення (див. рис. 16).

Детальна інформація про кімнату

Назва гуртожитку	Гуртожиток №1	Записаний на переселення
Номер кімнати	101	
Загальна кількість місць	3	
Кількість зайнятих місць	1	
Стать	Чоловіча	

Студенти, що проживають

Користувач	Дата поселення	Нотатки
Оліга Вєдєвіч	25.04.2021 20:04:43	Нотатки відсутні

Черга на поселення

Користувач	Дата, коли користувач став у чергу
Валентин Підгорний	26.04.2021 16:24:56
Свят Петренко	28.04.2021 16:30:16

Кнопка: Повернутись, повідомлення.

Рисунок 16 – Детальна інформацію про кімнату

5.3 Інструкція адміністратора

При успішній авторизації адміністратора, відкривається головна сторінка (див. рис. 17) та стають доступними всі функції керування даними на сайті. На панелі з'являються додаткові розділи для отримання всіх звернень до адміністрації та списку користувачів з відповідним позначенням підтвердження (див. рис. 18).

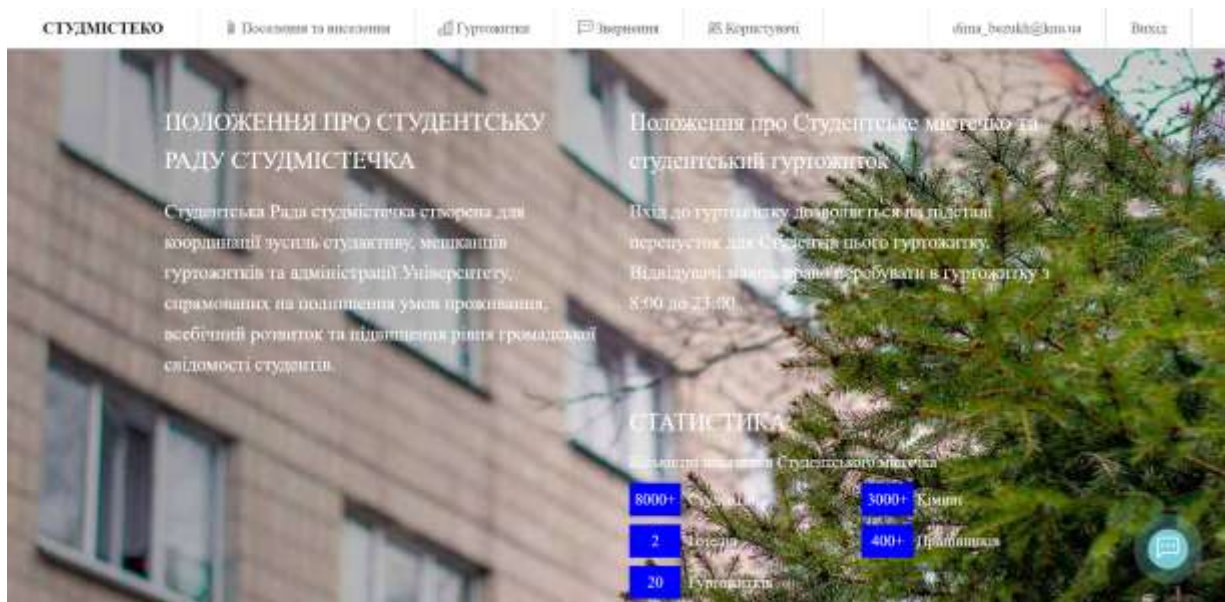


Рисунок 17 – Головна сторінка (вид для адміністратора)

Роль в системі	Ім'я	Прізвище	Електронна адреса	Дата створення облікового запису	Стать	Підтвердження	
Адміністратор	Ольга	Безух	olga_bezukh@knu.ua	01.05.2020	Жінка	Підтверджено	Переглянути деталі
Користувач	Олег	Петренко	oleg_petrov11@meta.ua	30.06.2022	Чоловік	Підтверджено	Переглянути деталі
Користувач	Валентин	Підгорний	vala@knu.ua	30.06.2022	Чоловік	Підтверджено	Переглянути деталі
Користувач	Лена	Шевченко	lena@knu.ua	30.06.2024	Жінка	Підтвердити користувача	Переглянути деталі

Рисунок 18 – Перелік користувачів

Перейшовши на сторінку не підтвердженого користувача, відображається його детальна інформація та можливість підтвердження його на роль адміністратора (див. рис. 19).

The screenshot shows the 'Електронний кабінет' (Electronic Cabinet) interface. At the top, there is a navigation bar with the following items: 'СТУДМІСТЕКО', 'Поселення та виселення', 'Гуртожитки', 'Звернення', 'Користувачі', 'anna_bezukh@knu.ua', and 'Вхід'. Below the navigation bar, the title 'Електронний кабінет' is centered. The main content area displays user information in a table format:

Ім'я	Анна
Прізвище	Шевченко
Електронна пошта	anna@knu.ua
Дата закінчення навчання	30.06.2024
Стать	Жіноча
Роль в системі	Користувач

Below the table, there is a button labeled 'Підтвердити як адміністратора'. On the right side of the interface, there is a blue chat icon.

Рисунок 19 – Підтвердження користувача як адміністратора

Перейшовши на сторінку звернення від користувача, відображається основний текст повідомлення та можливість відповісти на нього (див. рис. 20). Після того, як відповідь буде записана, користувачу, який надіслав звернення, приходить відповідний лист на пошту.

The screenshot shows the 'Інформація про повідомлення' (Message Information) interface. At the top, there is a navigation bar with the following items: 'СТУДМІСТЕКО', 'Поселення та виселення', 'Гуртожитки', 'Звернення', 'Користувачі', 'anna_bezukh@knu.ua', and 'Вхід'. Below the navigation bar, the title 'Інформація про повідомлення' is centered. The main content area displays message details in a table format:

Користувач	Олег Петренко
Тема повідомлення	Виселення
Повідомлення	Де можна знайти зразки документів для виселення?

Below the table, there is a section titled 'Відповісти на повідомлення користувача' (Respond to user message) with a large text input area. At the bottom left, there is a blue button labeled 'ПОВЕРНУТИСЯ' (Return) and a green button labeled 'ВІДПОВІСТИ' (Respond). On the right side of the interface, there is a blue chat icon.

Рисунок 20 – Форма для відповіді на питання

На сторінці детальної інформації про кімнату з'являється можливість керування чергою (підтвердження запитів на поселення) та виселення студентів, обов'язково вказавши причину (див. рис. 21).

СТУДМІСТЕКО | Поселення та виселення | Гуртожитки | Заявки | Користувачі | olga_bezukh@kma.ua | Вихід

Загальна кількість місць: 3
Кількість зайнятих місць: 2
Стать: Чоловіча

Студенти, що проживають

Користувач	Дата поселення	Нотатки
Мірошкін Казко	27.04.2021 15:41:10	нотатки відсутні
Степанів Вернатор	27.04.2021 15:42:38	нотатки відсутні

Черга на поселення

Користувач	Дата, коли користувач став у чергу
Олег Петренко	27.04.2021 16:13:18
Антон Чубренко	05.05.2021 14:05:33

Повернутись

Рисунок 21 – Детальна інформація про кімнату (вид для адміністратора)

На сторінці, де перелічені всі гуртожитки студентського містечка, з'являється можливість створення нового гуртожитку, натиснувши на іконку будівлі в лівому нижньому куті (див. рис. 22). На сторінці редагування інформації про певний гуртожиток, адміністратор може додати додаткове фото до опису гуртожитку та згенерувати кімнати, ввівши необхідні параметри в форму або створити кімнату вручну (див. рис. 23).

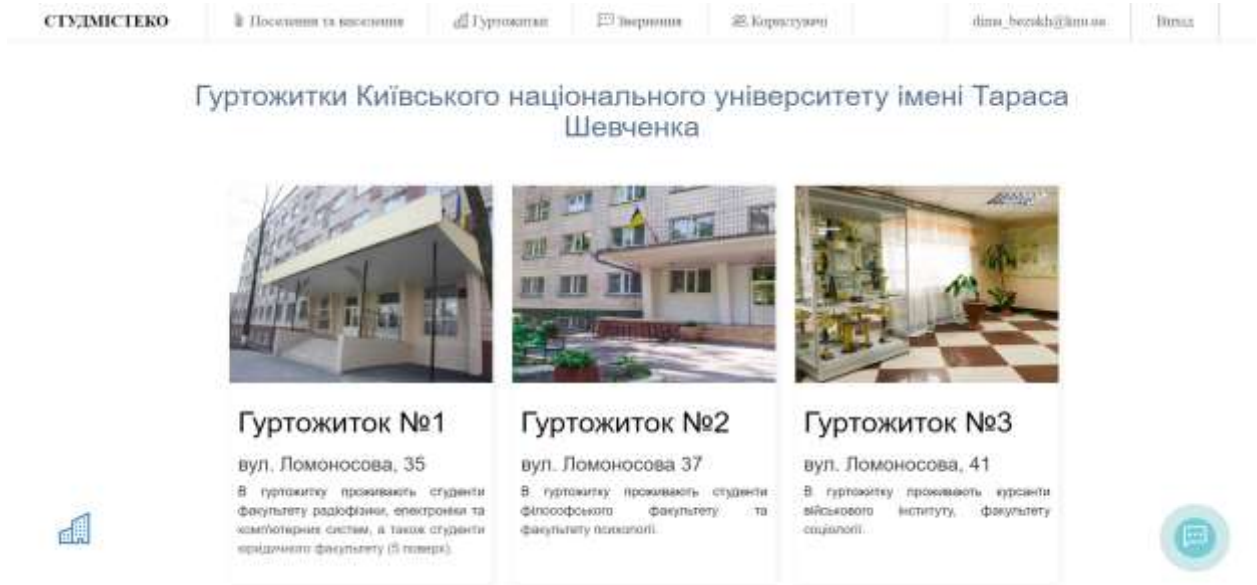


Рисунок 22 – Перелік гуртожитків студентського містечка (вид для адміністратора)

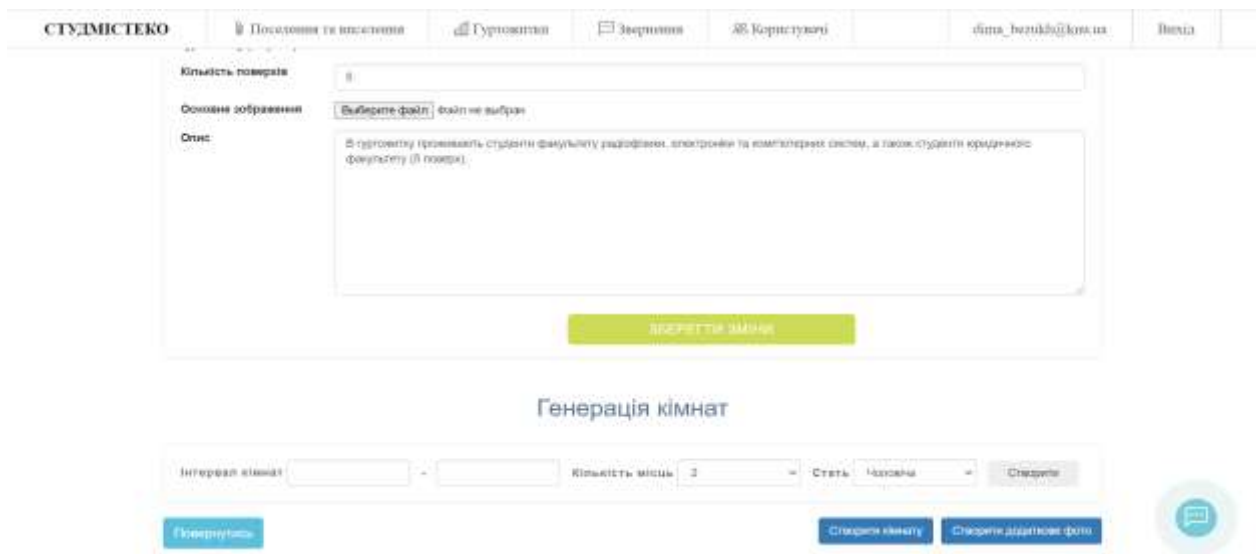


Рисунок 23 – Форма для редагування інформації про гуртожиток

ВИСНОВКИ

Під час розробки вебзастосунку досліджено діяльність студентського містечка Київського національного університету імені Тараса Шевченка. Проаналізовано існуючі додатки, які знайомлять користувача зі студентськими містечками закладів вищої освіти, визначено переваги та недоліки кожного з них, визначено аспекти для покращення та поставлено відповідні задачі.

Спираючись на поставлені задачі, було сформовано вимоги до функціоналу застосунку та побудовано модель бази даних, використовуючи яку, було розроблено серверну частину застосунку, функціонал якої розширювався, враховуючи думку замовника.

Розробляючи вебзастосунок, було прочитано необхідний обсяг документації, закріплено навички роботи з технологіями MVC, Razor та AJAX, взаємодією з базою даних та верстки інтернет-сторінок.

За підсумками розробки проєкту, можна відзначити, що результатом роботи є повністю працездатний вебзастосунок, який знайомить користувача зі студентським містечком університету, автоматизує роботу гуртожитків та відповідає всім поставленим вимогам.

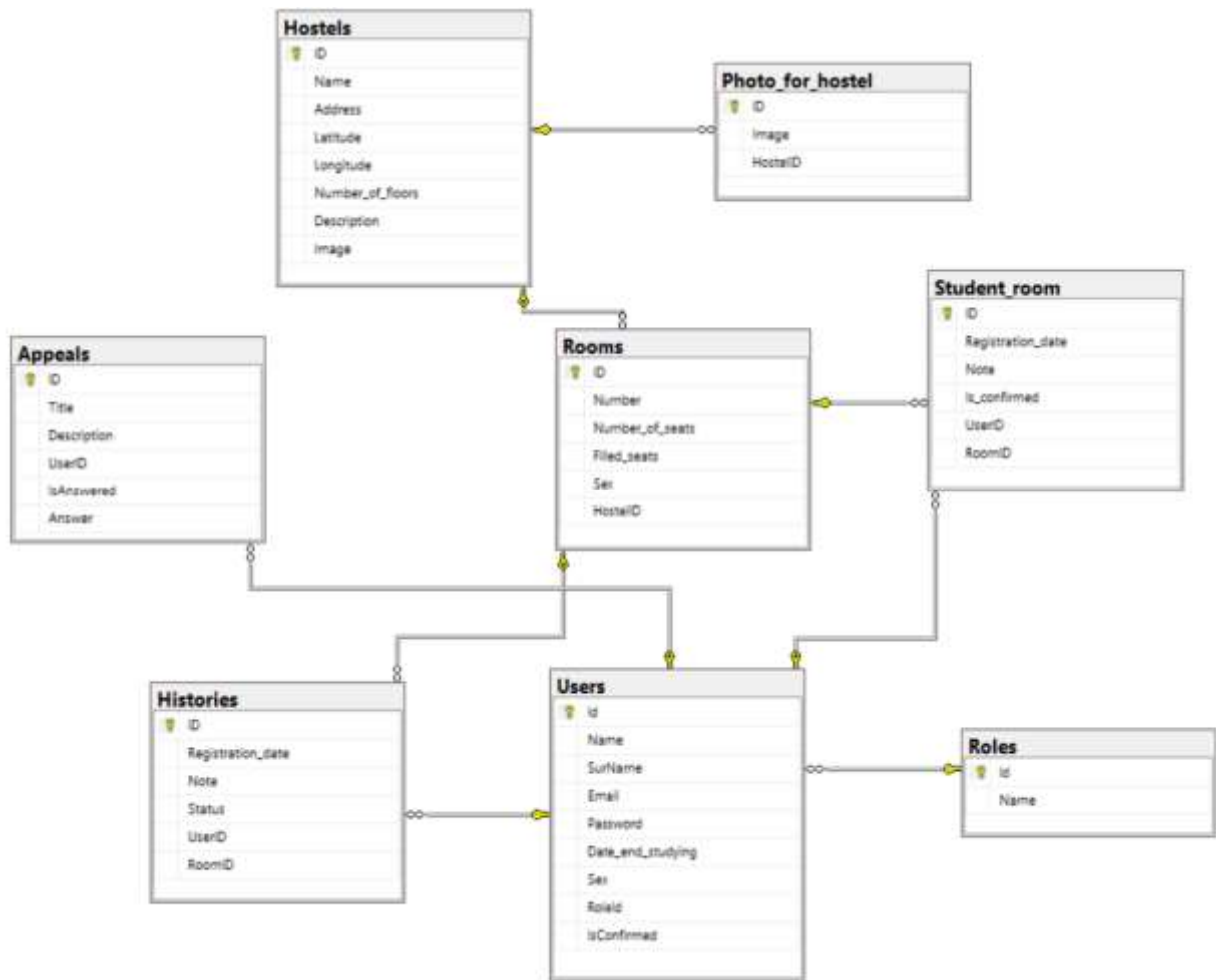
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Студмістечко Київського національного університету імені Тараса Шевченка [Електронний ресурс] – Режим доступу до ресурсу: <https://studmisto.knu.ua/>.
2. Студмістечко КПІ ім. Ігоря Сікорського [Електронний ресурс] – Режим доступу до ресурсу: <https://studmisto.kpi.ua/>.
3. Студентське містечко - ТНПУ [Електронний ресурс] – Режим доступу до ресурсу: <http://tnpu.edu.ua/about/studmist.php>.
4. Richter J. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Jeffrey Richter., 2013. – 896 с.
5. Адам Фримен. ASP.NET MVC 4 с примерами на C# 5.0 для профессионалов, 4-е издание / Адам Фримен – М.: «Вильямс», 2017. – 688 с.
6. Сандерсон С. ASP .NET MVC Framework с примерами на C# для профессионалов / Стивен Сандерсон., 2010. – 560 с.
7. Камерон Р. ASP.NET 3.5, компоненты AJAX и серверные элементы управления для профессионалов / Р. Камерон, Д. Михалк., 2008. – 608 с.
8. Закас Н. Ајах для профессионалов / Н. Закас, Д. Мак-Пик, Д. Фосетт., 2013. – 488 с.
9. Макдональд М. Microsoft ASP.NET 4 с примерами на C# 2010 для профессионалов / М. Макдональд, А. Фримен, М. Шпуста., 2011. – 1424 с.
10. ASP.NET [Електронний ресурс] – Режим доступу до ресурсу: <https://dotnet.microsoft.com/apps/aspnet>.
11. Introduction - C# language specification [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/language-specification/introduction>.

12. Полное руководство по языку программирования C# 9.0 и платформе .NET 5 [Электронный ресурс] – Режим доступа до ресурсу: <https://metanit.com/sharp/tutorial/>.
13. ASP.NET AJAX [Электронный ресурс]. – 2011. – Режим доступа до ресурсу: <https://docs.microsoft.com/en-us/aspnet/web-forms/videos/aspnet-ajax/>.
14. ASP.NET Razor [Электронный ресурс] – Режим доступа до ресурсу: https://www.w3schools.com/asp/razor_intro.asp.
15. BACINGER T. What is Bootstrap? A Short Bootstrap Tutorial on the What, Why, and How [Электронный ресурс] / TOMISLAV BACINGER – Режим доступа до ресурсу: <https://www.toptal.com/front-end/what-is-bootstrap-a-short-tutorial-on-the-what-why-and-how>.
16. Юдін А. ТОП БРАУЗЕРІВ ДЛЯ ПК, ПЛАНШЕТІВ, МОБІЛЬНИХ І ПРИСТАВОК В 2020 Источник: <https://marketer.ua/ua/stats-of-browsers/> [Электронный ресурс] / Антон Юдін. – 2020. – Режим доступа до ресурсу: <https://marketer.ua/ua/stats-of-browsers/>.

ДОДАТКИ

Додаток А. Діаграма бази даних



Додаток Б. Діаграма класів

