



**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА  
ДИЗАЙНУ**

**Факультет мехатроніки та комп'ютерних технологій**

**Кафедра комп'ютерні науки**

Рівень вищої освіти другий (магістерський )

Спеціальність 122 Комп'ютерні науки

Освітня програма Комп'ютерні науки

**ЗАТВЕРДЖУЮ**

**Завідувач кафедри комп'ютерні науки**

\_\_\_\_\_ Володимир ЩЕРБАНЬ

«\_\_\_\_\_» \_\_\_\_\_ 2023 \_\_\_\_\_ року

**З А В Д А Н Н Я**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТА**

**Каліненкову Владиславу Ігоровичу**

1. Тема роботи: Розроблення програмного продукту для автоматизації внутрішньо університетських процесів,

науковий керівник роботи: Астісова Тетяна Іванівна, к.т.н., доц.,

затверджені наказом закладу вищої освіти від 12.09.2023 року, № 210-уч.

2. Строк подання студентом роботи 12.11. 2023 р.

3. Вихідні дані до роботи: Розробка кафедри комп'ютерних наук

4. Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити)

РОЗДІЛ 1. Теоретичні аспекти предметної області; РОЗДІЛ 2. Аналіз систем та платформ для створення вебдодатків; РОЗДІЛ 3. Реалізація системи.

Додатки

5. Дата видачі завдання: 08. 2023р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапу кваліфікаційної роботи	Орієнтовний термін виконання	Примітка про виконання
1	Вступ	20.08.2023	
2	Розділ 1. Теоретичні аспекти предметної області.	10.09.2023	
3	Розділ 2. Аналіз систем та платформ для створення вебдодатків.	5.10.2023	
4	Розділ 3. Реалізація системи	25.10.2023	
5	Висновки	28.10.2023	
6	Оформлення кваліфікаційної роботи (чистовий варіант)	31.10.2023	
7	Подача кваліфікаційної роботи науковому керівнику для відгуку	01.11.2023	
8	Подача кваліфікаційної роботи для рецензування	04.11.2023	
9	Перевірка кваліфікаційної роботи на наявність ознак плагіату	06.11.2023	
10	Подання кваліфікаційної роботи на затвердження завідувачу кафедри	08.11.2023	

Студент

\_\_\_\_\_ Владислав КАЛІНЕНКОВ

Науковий керівник роботи

\_\_\_\_\_ Тетяна АСТІСТОВА

## АНОТАЦІЯ

**Каліненков В. І.** «Розробки програмного продукту для автоматизації внутрішньо університетських процесів».

Кваліфікаційна робота за спеціальністю 122 - «Комп'ютерні науки». – Київський національний університет технологій та дизайну, Київ, 2023 рік.

Кваліфікаційна магістерська кваліфікаційна робота складається з трьох розділів.

В першому розділі розкрита мета та завдання роботи вебсистеми діджиталізації у сфері освіти, а саме в гуртожитку; розглянуто види та особливості вебдодатків, показано їх переваги перед програмними аналогами; проведено аналіз переваг хмарних обчислень для поставленої задачі. В другому розділі проведено аналіз систем та платформ для створення вебдодатків та їх порівняльна характеристика. Третій розділ присвячено розробки функціональній моделі додатку; алгоритму роботи системи електронного поселення та тестування вебсистеми, огляду Google-продуктів; обранню мови програмування; баз даних .

Розробка дозволить перенести процеси, які існують у гуртожитку та в деканаті, у мобільний девайс чи персональний комп'ютер та може бути використана у будь-якому університеті чи іншій організації, яка потребує автоматизації процесу поселення людей у гуртожитки

Електронне поселення – це частина задачі діджиталізацію гуртожитку, яка вже нами реалізована та пройшла етап тестування на прикладі гуртожитку №3, факультету МКТ Київського національного університету технологій та дизайну та у Київському фаховому коледжі прикладних наук.

*Ключові слова: Sheets, Apps Script, Docs, Ajax, Google Cloud Platform технології, Microsoft Azure.*

## ANNOTATION

**Kalinenkova V. I.** "Development of a software product for the automation of intra-university processes."

Qualification work on specialty 122 - "Computer science". - Kyiv National University of Technology and Design, Kyiv, 2023.

The qualifying master's thesis consists of three sections.

In the first chapter, the purpose and tasks of the digitalization web system in the field of education, namely in the dormitory, are disclosed; types and features of web applications are considered, their advantages over software counterparts are shown; an analysis of the advantages of cloud computing for the given task was carried out. In the second section, an analysis of systems and platforms for creating web applications and their comparative characteristics is carried out. The third section is devoted to the development of the functional model of the application; algorithm of operation of the electronic settlement system and web system testing, review of Google products; choosing a programming language; databases.

The development will make it possible to transfer the processes that exist in the dormitory and in the dean's office to a mobile device or personal computer and can be used in any university or other organization that needs to automate the process of settling people in dormitories.

Electronic settlement is part of the task of digitizing the dormitory, which we have already implemented and passed the testing stage on the example of dormitory No. 3, Faculty of ICT of the Kyiv National University of Technology and Design and Kyiv College of Applied Sciences.

*Keywords: Sheets, Apps Script, Docs, Ajax, Google Cloud Platform технології, Microsoft Azure.*

## ЗМІСТ

<b>ВСТУП</b> .....	7
<b>РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ПРЕДМЕТНОЇ ОБЛАСТІ</b> .....	9
1.1. Постановка задачі.....	9
1.2. Web app .....	10
1.2.1.Поняття та особливості вебдодатка.....	10
1.2.2. Види вебдодатків.....	14
1.3. Хмарні технології .....	17
1.4. Принцип роботи вебдодатків та прикладного програмного забезпечення	20
1.4.1. Прикладне програмне забезпечення.....	21
1.4.2. Технологія веб-додатка.....	22
1.5. Різниця між веб-сайтом та веб-додатком.....	23
Висновки до першого розділу.....	26
<b>РОЗДІЛ 2. АНАЛІЗ СИСТЕМ ТА ПЛАТФОРМ ДЛЯ СТВОРЕННЯ</b>	
<b>ВЕБДОДАТКІВ</b> .....	27
2.1. Технологія Ajax.....	27
2.2. Google Cloud Platform .....	33
2.3. Microsoft Azure: Cloud Computing Platform & Services.....	37
Висновки до другого розділу.....	40
<b>РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ</b> .....	41
3.1. Огляд систем цифралізації гуртожитків.....	41
3.2. Огляд Google- продуктів.....	43
3.3. Мова для програмування в хмарі.....	45
3.4. Google Таблиці, як база даних.....	51
3.5 . Розробка діаграми послідовності.....	52
3.6. Алгоритм роботи системи та результати тестування .....	54
<b>ВИСНОВКИ</b> .....	61
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	62
<b>ДОДАТКИ</b> .....	

## ВСТУП

**Актуальність проблеми.** На сьогодні цифрові технології радикально змінюють природу інновацій в багатьох сферах діяльності. Цифрові інновації в освітньому процесі є частиною якісного навчання в усьому світі.

Для збереження конкурентоспроможності наших фахівців як в Україні так і за кордоном, проводиться цифралізація в освітніх процесах. Сучасний світ взяв напрям на діджиталізацію, значить і освітній процес буде все далі переходити в електронний формат. Цей процес є невідворотнім.

Сучасний навчальний процес у вищому навчальному закладі та сфери, які пов'язані з ним, все більше потребує використання цифрових та web-технологій.

Гуртожиток, як частина вищого навчального закладу, також потребує інновацій. Ідея про діджиталізацію гуртожитка виникла саме через не оптимізовані внутрішні процеси, які проходять в гуртожитку. Наша розробка допоможе вирішити одну із задач використання цифрових та web-технологій у сфері освіти, а саме в гуртожитку.

**Мета роботи.** Розробка програмного продукту для автоматизації внутрішньо - університетських процесів.

**Завдання роботи.** Завданням роботи було дослідити та проаналізувати існуючі інформаційні Інтернет-системи гуртожитків, нормативно – правову базу процесу поселення студентів; виділити місця, що потребують багато часу; провести тестування по зручності користування розробленим програмним додатком.

**Методика досліджень.** В основу методів дослідження розробки модуля управління гуртожитком було покладено ідею використання хмарних технологій для розробки вебдодатка електронного ресурсу. З використанням Google-продуктів.

**Об'єкт дослідження** - об'єктом дослідження виступає процес організації управління гуртожитками освітнього закладу.

**Предмет дослідження** – предметом дослідження є розробка системи для електронного поселення студентів у гуртожитки та ідентифікації студентів, що мешкають у гуртожитку.

**Наукова новизна.** Наукова новизна полягає в розробці системи електронного поселення студентів у гуртожиток на основі використання ідеї використання хмарних технологій.

Система призначена для локального використання на комп'ютері особи, відповідальної за облік студентів у гуртожитках, а для студентів, які мешкають у гуртожитку використовувати мобільні пристрої.

**Практичне значення отриманих результатів.** Результати тестування показали працездатність системи, можливість отримувати, аналізувати інформацію в реальному часі та оперативно реагувати на ситуації. Все це дасть можливість покращити життя студентів підняти працездатність працівників деканату та гуртожитку.

#### **Результати роботи були опубліковані в наступних виступах**

1. Астісова Т. І. Розробка WEB-додатку для автоматизації внутрішньо університетських процесів / Т. І. Астісова, В. І. Каліненков // Інформаційні технології в науці, виробництві та підприємстві : збірник наукових праць молодих вчених, аспірантів, магістрів кафедри комп'ютерних наук та технологій / за заг. наук. ред. В. Ю. Щербаня. – Київ : ТОВ "Фастбінд Україна", 2023. – С. 101-105.

<https://er.knutd.edu.ua/handle/123456789/24120>



## РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1. Постановка задачі.

Останні десятиліття цифрові технології все більше впливають майже на всі сфери діяльності та життя людини, як в роботі, так і дозвіллі.

З'являється поняття «цифрова людина», для якої цифрові технології стали невід'ємною частиною як у діяльності, так і в повсякденному житті.

В даний час розробляються цифрові технології наступного рівня, це рівень віртуальної та доповненої реальності, штучний інтелект, робототехніка, Інтернет речей, «розумні системи» тощо. Надалі можливе створення нейронних інтерфейсів та глибока інтеграція комп'ютера та людського мозку. Таким чином, діджиталізація змінює середовище освіти в університетах і змінює самі університети. Її вплив на людство буде не менш значним ніж вплив революції друку, що дозволило відтворювати та поширювати знання та ідеї у формі «друкованого слова». [1,2].

Зв'язок змін в університетах із загальною діджиталізацією суспільства та самого університету став загальноприйнятною ідеєю. Обговорюються різні компоненти діджиталізації університетів: 1) онлайн-освіта, 2) електронні освітні ресурси, 3) електронний документообіг, 4) навчання за допомогою тренажерів, віртуальної реальності, 5) формування компетенцій, необхідних у цифровому світі [2].

Цифрові технології, які глибоко проникають в саму суть продуктів багатьох організацій і радикально змінюють природу інновацій в продуктах та послугах. Цифрові інновації є частиною якісного навчання в усьому світі. Україні вже сьогодні варто інвестувати в ІТ - рішення для навчального процесу, щоб зберегти конкурентоспроможність на світовому ринку. [3].

Сучасний навчальний процес все більше і більше потребує втручання цифрових технологій. В Київському національному університеті технологій та дизайну було розроблено та успішно використовується електронний журнал викладача, який дає змогу вести облік студентів на парах та виставляти оцінки, за виконані завдання та платформа MOODLE яка використовується студентами

всіх форм навчання і особливо для дистанційної роботи студентів.

Заклади вищої освіти завжди зацікавлені в належному стані своїх гуртожитків та високому рівні умов життя своїх студентів. Наша розробка допоможе вирішити одну із задач діджиталізації у сфері освіти, а саме в гуртожитку.

Гуртожиток, як частина вищого навчального закладу, також потребує інновацій. Особливо оновлення потребує перебіг поселення в гуртожиток, який на даний момент не оптимізований і займає дуже багато часу. Ідея про діджиталізацію гуртожитка виникла саме через не оптимізовані внутрішні процеси гуртожитка, але у кінцевого додатку може перерости у дуже корисну соціально-освітню платформу [2].

Процеси поселення та обліку студентів у нашій державі не класифіковані і не стандартизовані – немає однозначних, або взагалі будь-яких спеціальних програмних засобів, що б допомагали зменшити витрати на паперові носії та дозволили б підвищити рівень інформаційного забезпечення закладів вищої освіти.

Розробка системи для автоматизації поселення та обліку студентів у гуртожитках університету є актуальною технічною задачею, вирішення якої приведе до покращення життя студентів.

Загалом, діджиталізація — це напрямок розвитку, який фактично є обов'язковим для більшості організацій. Він допомагає зміцнити зв'язки між адміністрацією, викладачами та студентами і сприяє кращому використанню наявних ресурсів. Але він вимагає від користувачів і організацій, щоб вони змінили світогляд.

## **1.2. Web app**

### **1.2.1. Поняття та особливості вебдодатків.**

Веб-додаток – розподілений додаток, в якому клієнтом виступає браузер, а сервером – веб-сервер [2].

Вебдодатки – це програми, написані скриптовою мовою (JavaScript, Perl, PHP та ін.) або написані мовою високого рівня та відкомпільовані під відповідну ОС (C, C++ та ін.), які працюють на стороні веб-сервера та призначені для створення інтерфейсу між користувачем та веб-сайтом.

Вебдодаток– це комп'ютерна програма, яка працює в браузері, зберігання та обробка інформації при такій організації обчислень відбувається на віддаленому сервері, а веб-переглядач служить програмою-клієнтом і призначеним для користувача інтерфейсом (рис. 1.1) [8].

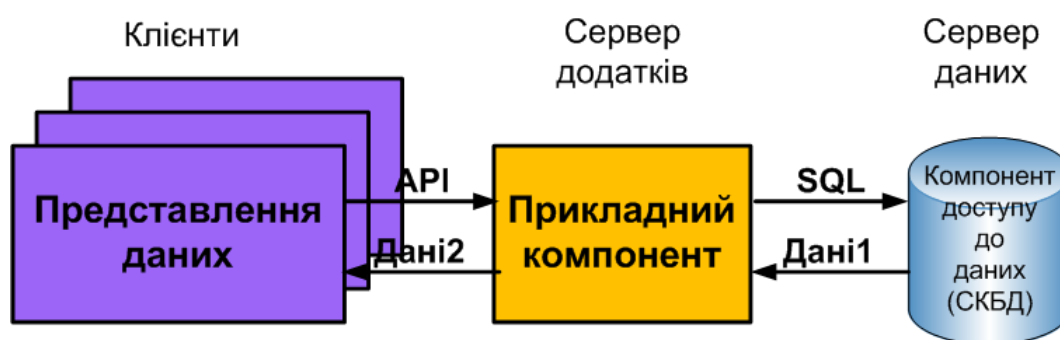


Рис.1.1. Схема роботи веб-додатку

Мобільний додаток існує на мобільному пристрої, а вебдодаток, це комп'ютерна програма, що виконує певну функцію

Вебдодаток може бути простим, як контактна форма на вебсайті, або складним, як текстовий процесор або багатокористувацький мобільний ігровий додаток, який завантажений на телефон.

Вебдодаток може працювати як текстовий процесор, зберігаючи інформацію в хмарі і дозволяти "завантажувати" документ на свій особистий жорсткий диск. Прикладом може бути Google Docs – це веб-програма, яка зберігає інформацію в хмарі і виконує функції текстового процесора.

Останнім часом з'явилися складні веб-програми які пов'язані з технологією AJAX. Вона є моделлю програмування для створення більш чуйних вебдодатків.

Workspace (раніше G Suite) та Microsoft Office 365,- це невеликий перелік

прикладів сучасних веб-додатків останнього покоління. Такі мобільні додатки: Facebook, додаток для онлайн-банкінгу, додаток Dropbox, які підключаються до Інтернету, були розроблені для все більш популярного використання мобільної мережі.

### *Особливості web-додатків.*

Веб-додаток отримує запит від клієнта, виконує обчислення, формує веб-сторінку та відправляє її клієнту. Для відправки через мережу використовується протокол HTTP.

Якщо бази даних розташованого на іншому сервері або іншому веб-додатку, то вебдодаток может бути клієнтом.

Оглянемо характеристики web-додатків:

#### 1. Незалежність від операційної системи клієнта

Додаток створюється один раз, але різна реалізація HTML, CSS, DOM в браузерах, викликає проблеми при розробці та подальшої підтримки. вебдодатків .

#### 2. Простота використання

Була відкинута теоретична побудова, присвячена гіпертексту, на користь простих прагматичних рішень. Ці рішення і послужили основою ідеальної конструкції.

REST— це архітектурний стиль для розподілених систем ( World Wide Web), який використовується для побудови веб-служб.

Системи, що підтримують REST, називаються RESTful-системами. REST є дуже простим інтерфейсом управління інформацією без використання додаткових внутрішніх прошарків, тобто передача даних відбувається в тому ж вигляді, що і самі дані. При розробки не потрібно загортати дані в XML, як це робить SOAP і XML-RPC, не використовувати AMF, як це робить Flash і т.д. Просто віддаємо самі дані. Кожна одиниця інформації визначається глобальним ідентифікатором - URL. Кожна URL в свою чергу має строго заданий формат

Якщо при розробки використовувати спрощені моделі програмування, то можна отримати вільно-зв'язаних партнерів. Проблема

корпоративних веб-сервісів в тому, що вони передбачають жорстко обумовлене партнерство. У багатьох випадках це виправдано, але частенько найцікавіші додатки можуть бути побудовані на вельми крихкій основі.

REST - займаються синдикацією даних, не контролюючи стан інформації на іншому кінці ланцюжка.

Однією з базових ідей самого Інтернету, є ідея кризної передачі даних.

### 3. Великий час реакції на дії користувача

Задача програміста - максимально скоротити час реакції додатка (до 10 секунд).

### 4. Велика кількість користувачів.

Десятки користувачів на секунду приходять на сайт успішних веб-проектів. Потрібно робити оптимізацію швидкості виконання програми.

### 5. Стійкість до атак.

Відкритий доступ до керування web-додатком дає змогу проводити хакерські атаки. Їх кількість на такі програми дуже велика.

### 6. Гігантські масиви даних

На сьогодні не існує інструментів для ефективного пошуку інформації в базах даних з мільярдами записів

### 7. Системи агентів, мережні, паралельні обчислення.

Додатки Internet часто спілкуються між собою. Так сайт "електронний магазин" використовує сайт банку, а розважальні сайти показують рекламні блоки з інших сайтів і т.д.

### 8. Програмне забезпечення – сервіс.

Головна характеристика сучасних Інтернет-додатків є те, що вони поширюються у вигляді сервісу, а це ведуть до фундаментальних змін в бізнес-моделях компаній-розробників програм.

Розрив між софтом-артефактом і софтом-сервісом спонукає підтримувати щодня розроблений продукт.

Google навчився автоматизувати процеси обслуговування мереж, системне адміністрування, що є ключовою частиною його цінової переваги перед

конкурентами. Google оновлює своє оточення в залежності від того, що новенького виявить користувач в своєму комп'ютерному оточенні сьогодні". Відстежування поведінки користувачів в реальному часі дозволяє бачити, які нові властивості використовуються і як вони використовуються - і це ще одна ключова складова успіху технології. Веб-розробник одного з розкритих мережесервісів відзначає: "ми додаємо дві-три нових властивості в різні частини сайту щодня, і якщо користувачам вони не подобаються - ми відмовляємося від цих нововведень. Якщо подобаються - упроваджуємо на всьому сайті"[7].

У прикладному відношенні – вебдодатки мають істотну перевагу: вже розроблено багато програм і сервісів, за допомогою яких будь-яка людина, може створювати різні корисні програми для своєї зручності та розваги. Причому абсолютно безкоштовно.

Наприклад, відомий Gmail є повноцінним поштовим клієнтом, який запускається на вашому комп'ютері, має безліч додаткових функцій, робить все, що робить будь-який інший поштовий клієнт. Bloglines – вебдодаток для читання новин, який конкурує із звичайними аналогічними застосуваннями, конкурує і виграє.

Ці вебдодатки працюють на сервері, а їх користувацький інтерфейс (UI) відображається у вигляді веб-сторінок.

### **1.2.2. Види вебдодатків.**

Існує декілька видів вебдодатків, зазвичай їх поділяють на три типи :



Рисунок 1.2- Вид вебдодатків

**1). Native mobile apps** - мобільні застосунки, знаходяться безпосередньо на мобільному телефоні.

Магазини додатків Google Play та App Store потрібні цим застосункам для завантаження на пристрій. Для розробки використовують мови програмування: Java, Kotlin, Swift, Python, C # тощо.

## 2). Hybrid app

Гібридні додатки працюють як вебпрограми через браузер, мають свій ярлик на домашньому екрані, чуттєвий дизайн та високу продуктивність.

Гібридні додатки, не прив'язуються до платформи. Це вебзастосунки, які мають вигляд як нативні. При розробці використовують нативні і веб-технології. Різниця в використанні технологій показано на рис.1.3.



Рисунок 1.3- Технології Native та Hybrid застосунків

Цей вид додатку швидко завантажується, забезпечує постійний користувацький інтерфейс та інтеграцію з файловими системами мобільних пристроїв і потребує набагато менше коду. Використовує технології HTML, CSS, JavaScript тощо.

Hybrid app дозволяє розробникам написати коди один раз і відкривати їх на різних платформах. Механізм браузера пристрою використовується для відображення HTML та JavaScript та власних API-інтерфейсів для доступу до апаратного забезпечення конкретного пристрою.

Таким застосункам може не вистачати потужності та швидкості, що є відмінною характеристикою нативних застосунків.

### 3) Web app або вебдодаток.

Вебзастосунки –це скоріше вебсайти, які не треба завантажувати на пристрій. Доступ до них відбувається через веббраузери. Вони виглядають як звичайні веб-сайти Вони можуть адаптуватись до пристрою через який користувач зайшов ( рис. .1.4).

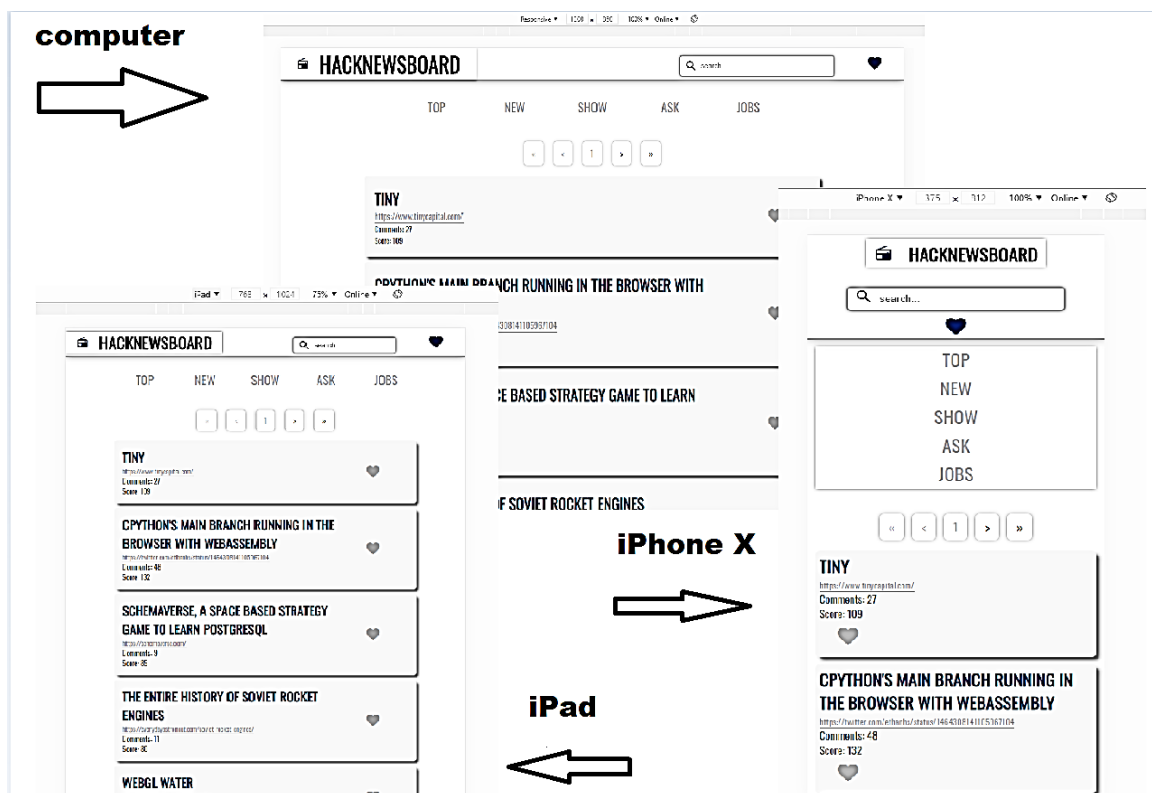


Рисунок 1.4 - Адаптивність на прикладі вебдодатку



Для розробки цих додатків використовується HTML, Javascript, CSS тощо. Головною вимогою для роботи, це завантажений веббраузер.

Вебдодатки мають низьку вартість розробки, їх достатньо легко обслуговувати, для завантаження оновлень не потрібно перезавантажувати програму

Вебпрограми розробляють для взаємодії з користувачами. Вони здатні обробляти інформацію, яку вносить користувач. На рисунку 1.5 показана різниця між вебсайтами та вебдодатком.

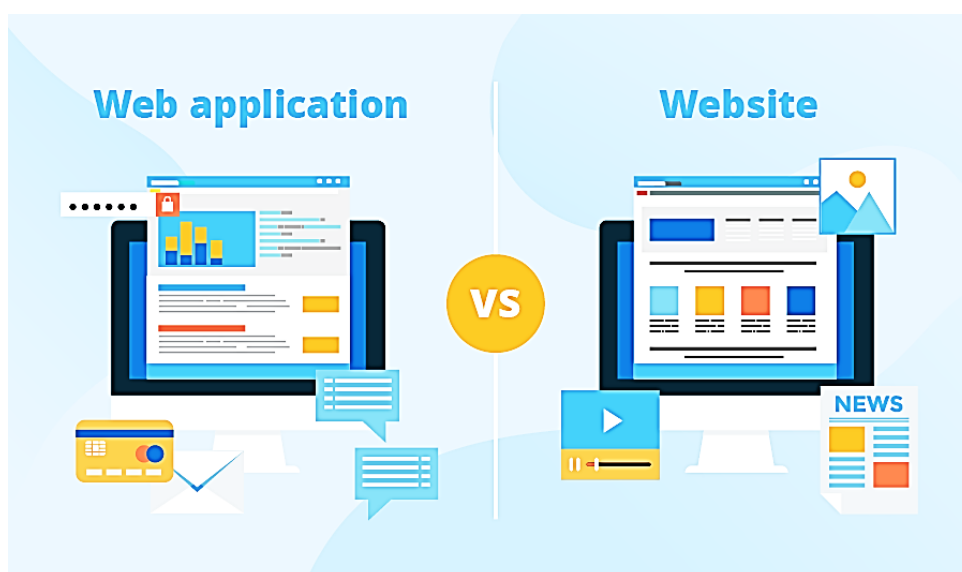


Рисунок 1.5.- Вебдодаток та вебсайт

Вебзастосунки надають користувачу можливість взаємодіяти з інтерфейсом й отримувати відповідь, їх не потрібно налаштовувати під різні платформи.

Вебзастосунок можливо відкрити на мобільному пристрої з браузером та на комп'ютері, планшеті, лаптоці.

Для створення вебдодатку типово використовувати JavaScript.

### 1.3. Хмарні технології

Хмарні технології – це парадигма, що передбачає віддалену обробку та зберігання даних [6]. Уперше про технології заговорили у 2008 році.

Хмара – це деякий ЦОД (дата-центр, сервер) або їх мережа, де

зберігаються дані та програми, що з'єднуються з користувачами через Інтернет (рис. 1.6).

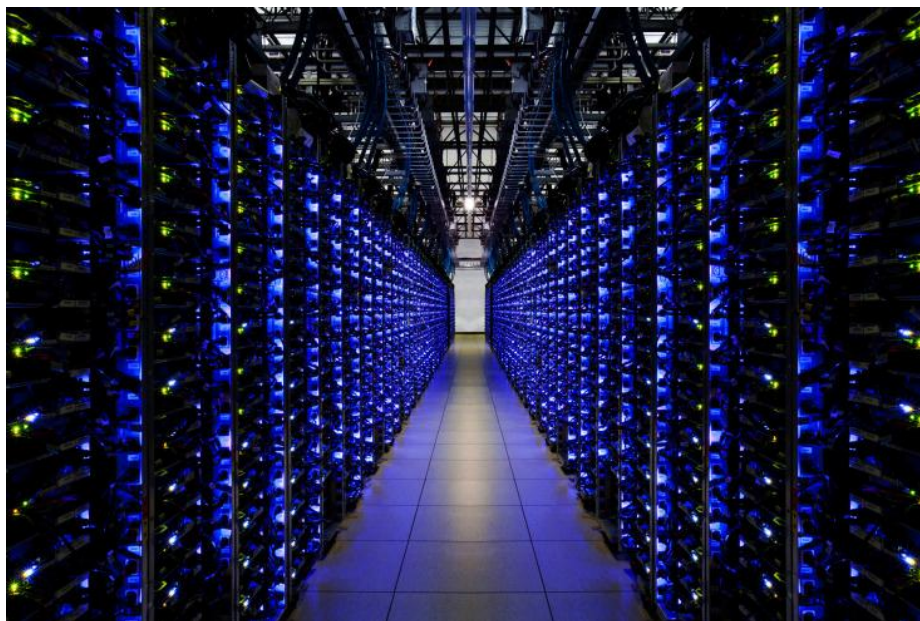


Рисунок 1.6.- Дата центр

Хмарні технології – це одна велика концепція, що включає в себе багато різних понять. Це і програмне забезпечення, інфраструктура, платформа, дані, робоче місце і т.п. Найголовнішою функцією хмарних технологій є задоволення потреб користувачів, що потребують віддаленої обробки даних.

Хмарні технології дозволяють використовувати програми без установки їх на робочу машину і надають доступ до особистих файлів з будь-якого комп'ютера, що має доступ в Інтернет. Ця технологія дозволяє значно ефективніше керувати, обробляти та працювати з інформацією за рахунок централізації цієї самої інформації.

Прикладом хмарних технологій є Gmail, Meta і т.д. Щоб відправити пошту, потрібно лише підключення до Інтернету. Зауважимо, що додаткового програмного забезпечення або сервера не потребується.

Хмарні технології, - це постачання послуги хостингу через Інтернет.

Хостинг – це послуга по розміщенню обладнання клієнта на території провайдера, при цьому забезпечується підключення його до каналів зв'язку з високою пропускнуною спроможністю [8].

В хмарних технологіях комп'ютерні ресурси надаються Інтернет-користувачу як онлайн сервіси.

Ці хмарні сервіси ( Інтернет послуги) можна розділити на три основні категорії:

- програмне забезпечення як сервіс (SaaS);
- платформа як сервіс (PaaS);
- інфраструктура як сервіс (IaaS).

### **1. Програмне забезпечення як сервіс (SaaS).**

По моделі SaaS постачається апаратна інфраструктура і програмне забезпечення та взаємодію з користувачем через інтерфейсний портал.

По моделі SaaS можуть надаватись самі різноманітні послуги, від веб-пошти до управління запасами, обробки БД. Перевагою моделі- користувач може користуватись послугою з будь-якої точки світу.

### **2. Платформа як сервіс (PaaS).**

PaaS- це набір засобів розробки та програмних продуктів, що розміщені на інфраструктурі провайдера. Розробники на платформі провайдера через Інтернет можуть створювати свої програми.. PaaS провайдери використовують шлюзи, сайт-портали, API, або програмне забезпечення установлене на комп'ютері клієнта.

### **3. Інфраструктура як сервіс (IaaS).**

IaaS являє собою віртуальний сервер instanceAPI для запуску, зупинки, доступу, налаштування своїх віртуальних серверів та систем збереження. IaaS дозволяє компанії платити саме за стільки потужностей, скільки їй необхідно. Дану модель іноді називають «комунальні обчислення» [5].

У порівнянні з традиційним підходом, хмарні сервіси дозволяють керувати значними інфраструктурами, обслуговувати різноманітні групи користувачів в межах однієї хмари, а також означають повну залежність від провайдера хмарних послуг.

Переваги хмарних технологій:

- користувач сплачує за послугу тільки тоді, коли вона необхідна; сплачує лише за те, що використовує;
- дозволяють економити на придбанні, підтримці, модернізації ПО та обладнання;
- масштабність, відмовостійкість та безпечність
- автоматичне виділення необхідних ресурсів в залежності від потреби;
- технічне обслуговування, оновлення ПО проводить провайдер послуг.

Недоліки хмарних технологій:

- збереження інформації користувача залежить від компанії провайдера;
- необхідно мати якісний Інтернет;
- є ризик, що провайдер не зробить резервну копію даних, і вони будуть втрачені;
- надаючи свою інформацію онлайн
- сервісу, користувач втрачає над ними -є ризик, що провайдер не зробить резервну копію даних, і вони будуть втрачені;
- надаючи свою інформацію онлайн
- сервісу, користувач втрачає над ними контроль і обмежує свою свободу.

#### **1.4. Принцип роботи веб-додатків та прикладного програмного забезпечення.**

##### **1.4.1. Прикладне програмне забезпечення**

В основу роботи комп'ютерів покладено програмний принцип керування. Комп'ютер виконує дії за заздальгідь заданою програмою. **Програма** – це запис алгоритму розв'язання задачі у вигляді послідовності команд або операторів мовою, яку розуміє комп'ютер. Кінцевою метою будь-якої комп'ютерної програми є керування апаратними засобами [9].

Для нормального розв'язання задач на комп'ютері потрібно, щоб програма була налагоджена і мала відповідну документацію. Термін програмне забезпечення (software), це сукупність програм, процедур, правил та документації, що стосуються функціонування системи оброблення даних.

Склад програмного забезпечення обчислювальної системи називається програмною конфігурацією. Між програмами існує взаємозв'язок у вигляді піраміди, де кожен вищий рівень базується на програмному забезпеченні попередніх рівнів (рис. 1.7).

Прикладний рівень
Службовий рівень
Системний рівень
Базовий рівень

Рисунок 1.7 Структура програмного забезпечення

#### *Базовий рівень*

Відповідає за взаємодію з базовими апаратними засобами, утворюючи базову систему введення-виведення BIOS. Програми та дані записуються у ПЗП на етапі виробництва і не можуть бути змінені в процесі експлуатації.

#### *Системний рівень*

Програми цього рівня забезпечують взаємодію інших програм комп'ютера з програмами базового рівня і безпосередньо з апаратним забезпеченням. Від програм цього рівня залежать експлуатаційні показники всієї обчислювальної системи. Конкретні програми, призначені для взаємодії з конкретними пристроями, називають драйверами. Сукупність програмного забезпечення системного рівня утворює ядро операційної системи комп'ютера яке надає можливості практичної роботи користувача з обчислювальною системою. Ядро операційної системи виконує такі функції: керування

пам'яттю, процесами введення-виведення, файловою системою, організація взаємодії та диспетчеризація процесів, облік використання ресурсів, оброблення команд і т.д.

#### *Службовий рівень*

Призначення службових програм (утиліт) полягає у автоматизації робіт по перевірці та налаштуванню комп'ютерної системи, а також для покращення функцій системних програм.

### **1.4.2. Технологія веб-додатка**

Розглядаючи структуру додатку, бачимо, що вся програмна логіка сконцентрована на сервері, а користувацький інтерфейс, як невелика програма, доступний будь-якій людині. Ця програма існує з самого моменту зародження HTTP-мережі ( маємо на увазі браузер). Для появи веб-додатків не вистачало тільки таких важливих технологій, як JavaScript і DOM.

Технологія JavaScript в 1999 р. використовувалась класним способом - змусити елементи сторінки реагувати на рух миші (що зараз всіх дратує), створювати сліди курсору на екрані, блимати текстом, міняти кольори.

Сьогодні ж JavaScript – це повноцінна мова використовується для виконання дій на стороні клієнта без зайвого звернення до віддаленого серверу.

Динамічна складова веб-сторінок, яка завжди була слабкою частиною при створенні веб-додатків, сьогоднішній JavaScript повністю вирішує

Отже, вся програмна логіка програми знаходиться на сервері, на відміну від звичайного ПЗ, де логіка програми розташовується на комп'ютері кожного користувача.

Проблеми поширення веб-додатку не існує, оскільки отримати ви його можете в будь-який момент в будь-якому місці.

Для роботи користувачеві потрібна тільки одна програма і це браузер, набравши URL можете приступати до роботи.

Якщо користувач встановлює на своїй машині програму, то йому доводиться виконувати роль адміністратора, а саме : встановлювати програму, запускати, налаштовувати, лагодити, вирішувати проблеми, що виникають.

У випадку ж з веб-додатком користувачеві немає необхідності турбуватися, тому що в ролі адміністратора виступає розробник додатку. Веб-додаток не пред'являє ніяких вимог до апаратної платформи. Він зможе працювати з додатком на будь-якій операційній системі і не помітить ніякої різниці. Не існує проблеми підтримки різних версій. Як тільки виходить нова версія веб-додатку, достатньо просто оновити сторінку в браузері.

Використання пам'яті, не турбує користувача, тому що досить завантажити тільки ту його частину, яка потрібна для виконання конкретної поточної задачі. Навіть весь інтерфейс не обов'язково завантажувати. Досить завантажити тільки ту його частину, яка потрібна для виконання конкретної поточної задачі. Завдяки цьому веб-додатки завантажуються всього за кілька секунд

На комп'ютері користувача нічого не встановлюється, тому він може працювати з додатком з будь-якого місця на Землі.

Ніяких проблем із закритими протоколами і номерами портів.

Архітектура додатка невидима для користувача і він навіть не підозрює про те, як все влаштовано на стороні сервера.

### **1.5. Різниця між веб-сайтом та веб-додатком**

Сайт необов'язково є веб-додатком і створюється в інформаційних цілях. Сайти можуть бути статичними або динамічними, але основна мета – це надання інформації [11].

*Статичні сайти*

*Веб-додатки є інтерактивними*

Особлива риса, яка відрізняє вебдодаток від вебсайта це інтерактивність. Це означає, що користувач активний учасник при роботі з

сайтом. Користувач вебдодатку шукає необхідну інформацію, заповнює форми, виконує платежі через особистий кабінет і т.д.

За складністю виконання подібних рішень веб-додатки істотно складніше інформаційних сайтів, а відповідно дорожче. Наприклад, Amazon.com (рис. 1.8). Погляньте на домашню сторінку сайту Amazon.com – це гібрид, який однак, є інформаційним сайтом.

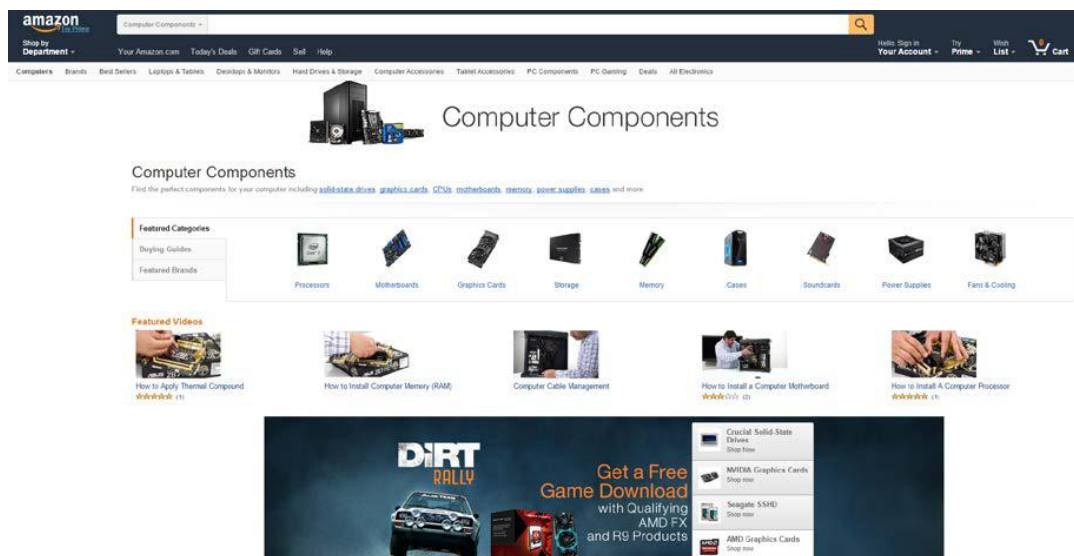


Рисунок 1.8 Amazon.com

На скріншоті (рис 1.8.) представлено багато інформації. Процесори, материнські плати, пам'ять, ціни і категорії – все це види інформації. Але головною метою користувача є взаємодія. Користувач може виконати пошук або подивитися, що знаходиться в кошику; перейти в якусь категорію товарів; детально ознайомитися з яким-небудь товаром і купити його. Все це види взаємодій.

Веб-сайт представляє яку-небудь інформацію, то потрібно визначити баланс між інформацією та інтерактивністю. Якщо відтворювати багато взаємодій, то це інтерактивний сайт.

Знаючи всю цю інформацію про сайти, можна зробити висновок: що вебдодатком є певний сайт, а саме динамічний, який включає в себе максимум інтерактивності, меншу частину відведено на інформативну складову, звичайно можуть бути і виключення, на прикладі розважальних



інформаційних сайтів, які надають якусь цікаву інформацію і при цьому надають користувачу інтерактивні можливості в опрацюванні цієї ж інформації

## **Висновки до першого розділу.**

На основі цього можна зробити висновки: веб-додаток є актуальним в наш час. Ця технологія стрімко розвивається і покликана економити час та ресурси на створенні універсальних додатків (програм), які в свою чергу дають можливість працювати в будь-якому місці, у будь-який час та з будь-яких девайсів не залежно від операційної системи чи технічних характеристик.

Хоча веб-додатки не зможуть робити деякі речі, які доступні настільним додаткам, та про це мова і не йде. Ви ніколи не зможете з браузера створювати складні тривимірні моделі. Веб-додатки не можуть ніяк замінити системні програми. Не зможуть принаймні в найближчому майбутньому.

Але не зважаючи на це, веб-додатки – технологія майбутнього. Вони починають пробиватися до світла і замінювати собою програмні засоби якими користувач користується в повсякденному житті. На нашу думку, веб-додатки – це майбутнє Інтернет мережі, програмних додатків, мережевих архітектур і навіть комп'ютерів і операційних систем. Веб-додатки проникнуть мало не у всі види програм, якими ми користуємося сьогодні.

В підсумку можна сказати, що веб-додатки мають явні переваги перед своїми програмними аналогами в основному це їх мобільність, простота в створенні та використанні, тому ця технологія стрімко набуває популярності, як серед користувачів, так і серед розробників.

Знаючи всю цю інформацію про сайти, можна зробити висновок: що веб-додатком являється певний сайт, а саме динамічний, який включає в себе максимум інтерактивності, меншу частину відведено на інформативну складову, звичайно можуть бути і виключення, на прикладі розважальних інформаційних сайтів, які надають якусь цікаву інформацію і при цьому надають користувачу інтерактивні можливості в опрацюванні цієї ж інформації

## РОЗДІЛ 2. АНАЛІЗ СИСТЕМ ТА ПЛАТФОРМ ДЛЯ СТВОРЕННЯ ВЕБ-ДОДАТКІВ

### 2.1. Технології Ajax

Технології Ajax (рис. 2.1) є базовою в створенні веб-додатків

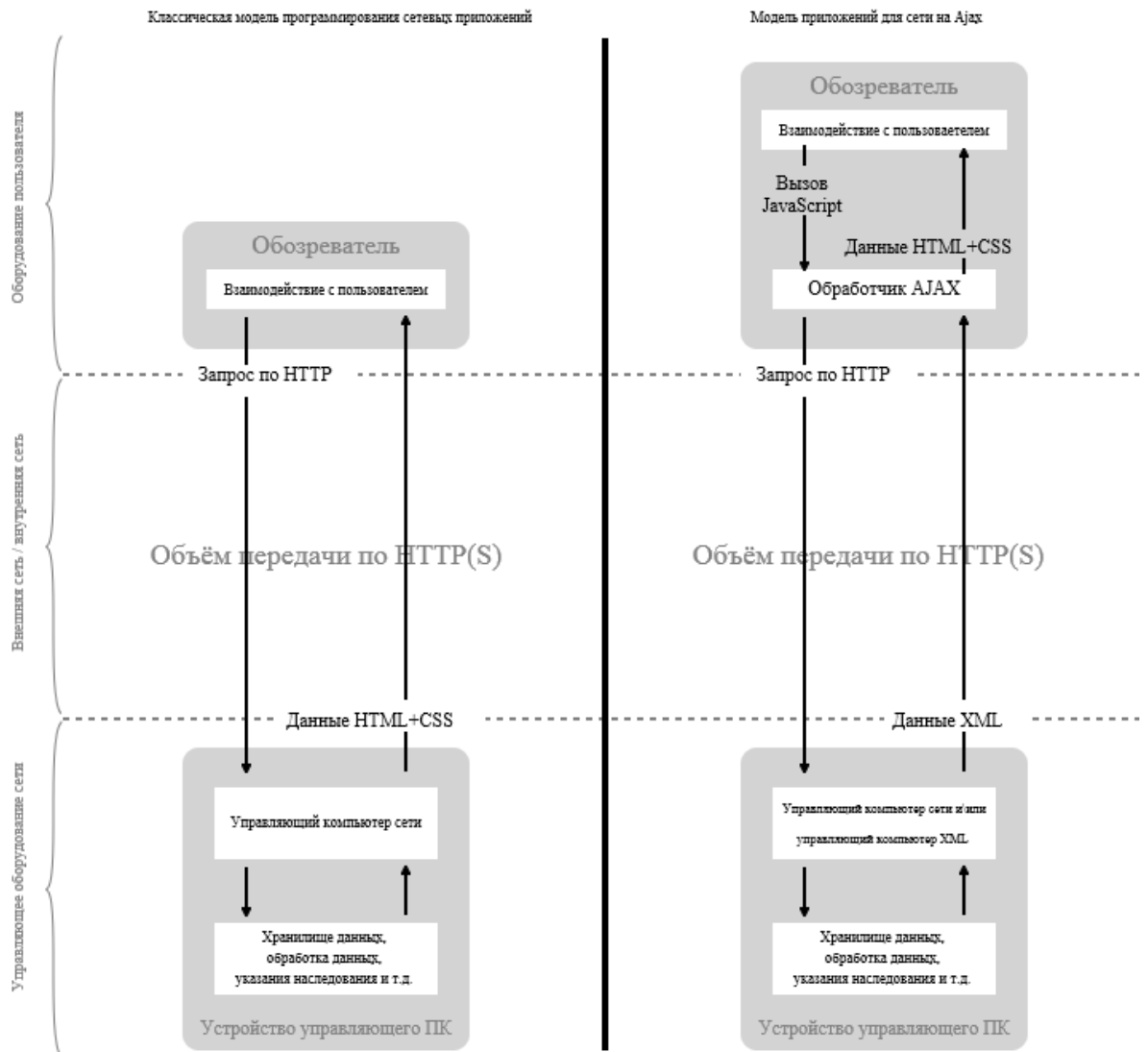


Рисунок 2.1- Схема роботи технології Ajax

AJAX (Asynchronous JavaScript And XML) – технологія побудови користувацьких інтерфейсів веб-додатків. Ця технологія дозволяє не перезавантажувати веб-сторінки, а у фоновому режимі надсилати запити на

сервер і довантажувати потрібні користувачу дані. AJAX є одним з компонентів концепції DHTML.

Ця технологія з'явилась в лютому 2005 р. у статті Джесі Джеймса Гарретта (Jesse James Garrett) «Новий підхід до веб-додатків». Проте в тій чи іншій формі багато технологій були доступні і використовувалися набагато раніше, наприклад в підході «Remote Scripting», запропонованому компанією Microsoft в 1998 р., або з використанням HTML-елемента IFRAME, що з'явився в Internet Explorer 3 в 1996 р.

AJAX – це концепція декількох суміжних технологій, що комбінуює кілька основних методів і прийомів:

- використання DHTML для динамічної зміни змісту сторінки;
- використання XMLHttpRequest для звернення до сервера «на льоту», не перезавантажуючи всю сторінку повністю;
- альтернативний метод – динамічне підвантаження коду JavaScript в тег <SCRIPT> з використанням DOM, що здійснюється із використанням формату JSON;
- динамічне створення дочірніх фреймів.

Використання цих підходів дозволяє створювати набагато зручніші веб-інтерфейси користувача на тих сторінках сайтів, де необхідна активна взаємодія з користувачем.

AJAX – асинхронний і користувач може переглядати контент сайту, поки сервер ще обробляє запит. Браузер не перезавантажує веб-сторінку і дані посилаються на сервер без візуального підтвердження.

Компанія Google почала активно використовувати AJAX при створенні таких сайтів, як Gmail, Google Maps і Google Suggesti і він отримав велику популярність. Створення цих сайтів підтвердило ефективність використання даного підходу.

*Класична модель веб-додатку:*

1. Користувач заходить на веб-сторінку і натискає на який-небудь її елемент.

2. Браузер надсилає запит серверу.
3. Сервер (у відповідь ) генерує повністю нову веб-сторінку і відправляє її браузеру і т. д.
4. Сервер може генерувати не всю сторінку наново, а тільки деяких її частин.
5. Подальша передача сторінки користувачу.

#### *Модель AJAX:*

1. Користувач заходить на веб-сторінку і натискає на який-небудь її елемент.
2. Браузер відправляє запит на сервер.
3. Сервер віддає тільки ту частину документа, яка змінилася.

В деяких додатках використовуються певні варіації з форматом відповіді сервера. Ці варіації набули напівофіційних назв.

АНАН (Asynchronous HTML and HTTP) – це споріднений AJAX підхід для динамічного оновлення веб-сторінок, з використанням JavaScript. Основна його відмінність в тому, що відповіді сервера повинні бути звичайним HTML. Перевагою підходу є більше сумісності і функціональності, а саме: підтримка навігаційних кнопок браузера, аплоад файлів тощо.

Реалізується АНАН у вигляді звичайних фреймів, які автоматично змінюють свій розмір під розмір вмісту, або у вигляді прихованих фреймів, що виконують тільки функції завантаження даних.

Asynchronous XHTML and HTTP, або аббревіатура АХАН – це майже те ж саме що і АНАН. Різниця тільки в тому, що в АНАН сервер клієнтові повертає HTML, а в АХАН вже XHTML.

Використання DHTML (англ. Dynamic HTML) для динамічної зміни вмісту сторінки, складається з CSS, DOM і JavaScript. Фрагменти простого тексту, HTML-коду, JSON або XML можна використовувати в якості формату передачі даних.

- a) DOM (англ. Document Object Model),

Дії користувача з інтерфейсом перетворюються в операції з елементами DOM. За допомогою цих операцій обробляються дані, доступні користувачеві, в результаті чого подання їх змінюється. Також проводиться обробка переміщень і клацань мишею, натискань клавіш.

б) CSS (англ. Cascading Style Sheets)

Каскадні таблиці стилів забезпечують узгоджений зовнішній вигляд елементів програми та спрощують звернення до DOM-об'єктів.

в) Об'єкт XMLHttpRequest (подібні механізми)

Подібні механізми використовується для асинхронного взаємодії з сервером, обробки запитів користувача і завантаження в процесі роботи необхідних даних.

Розглянемо переваги та недоліки технології AJAX.

**Переваги технології AJAX.**

- *Економія трафіку.*

При роботі з веб-додатком замість завантаження всієї сторінки достатньо завантажити тільки частину, що змінилася (скорочується трафік).

- *Зменшення навантаження на сервер.*

AJAX дозволяє в кілька разів знизити навантаження на сервер. Всі сторінки сайту найчастіше генеруються за одним шаблоном («шапка», «навігаційна панель», «підвал» і т.д.). Для їх генерації потрібні звернення до різних файлів. Якщо генерується та передається лише змістовна частина сторінки, то час на обробку скриптів, запитів до баз даних можна опустити. Дизайн сторінки також містить безліч файлів, пов'язаних з оформленням (картинки, стилі). Використовуючи AJAX не треба витратити часу на повторну обробку сторінок.

- *Прискорення реакції інтерфейсу.*

Завантаження частини сторінки, що змінилася проходить значно швидше, то користувач бачить результат своїх дій швидше.

- *Безмежні можливості для інтерактивної обробки.*

При введенні пошукового запиту в Google виводиться підказка з можливими варіантами запиту (при реєстрації на сайтах, користувач вводить ім'я, і відразу бачить, чи доступне воно). AJAX зручний для програмування чатів, адміністративних панелей та інших інструментів, які виводять мінливі з часом дані.

### **Недоліки технології AJAX.**

- *Відсутність інтеграції зі стандартними інструментами браузера.*

Браузер не реєструє динамічно створювані сторінки в історії відвідування сторінок (кнопка «Назад», не працює). Для вирішення цієї проблеми існують скрипти.

- *Заміна вмісту сторінки при постійному URL.*

Заміна сторінок призводить до неможливості збереження закладки на бажаний матеріал. History.pushState успішно вирішує цю проблему

- *Проблеми для пошуковиків.*

Пошуковикам не доступно динамічно завантажувати сторінки. Вирішити цю проблему можна, якщо не перевіряти запит, звичайний він або XMLHttpRequest.

Пошукові машини не можуть виконувати JavaScript. Розробникам потрібно подбати про альтернативні способи доступу до вмісту сайту.

- *Неактуальність старих методів обліку статистики сайтів*

На сьогодні вже є багато сервісів, які ведуть облік переглядів нових сторінок сайту. Для сайтів, що використовують AJAX, така статистика вже стає неактуальною.

- *Ускладнення проєкту.*

Логіка обробки даних перерозподіляється. На сторону клієнта частково перенесуться процеси первинного форматування даних. Це ускладнює контроль цілісності форматів і типів. Зростають витрати на кодування і управління проєктом, а також ризиком зниження доступності сервісу..

- *JavaScript потрібен бути включений в браузері.*

Якщо з міркувань безпеки JavaScript може бути вимкнений, то AJAX-сторінки будуть важкодоступні неповно функціональним браузерам, веб-архівам та роботам.

- *Низька швидкість при грубому програмуванні.*

Призначенням AJAX є підвищення швидкості. Але, коли AJAX-запитів на одній сторінці багато (по кожному кліку довантажуються список), AJAX-сторінка стає повільнішою традиційної.

- *Ризик фабрикації запитів іншими сайтами.*

Тег `<script>`, на відмінність від XMLHttpRequest, не діє в межах одного документу. Результат роботи AJAX-запиту може бути JavaScript-кодом (зокрема, JSON).

Наприклад, якщо написати код

```
<Script type = "text / javascript" src = "http://example.org/inbox.php"> </script> ,
```

то в HTML буде вставлена папка вхідних повідомлень того користувача, який на цьому комп'ютері авторизований на example.org. Для захисту від цього використовують POST-запит. Але GET вважається ідемпотентність і тому кешується, POST – ні. Для вирішення цієї проблеми, Google вставляє в початок відповіді нескінченний цикл, при цьому AJAX може робити з відповіддю що завгодно (прибрати цикл), а тег `<script>` підключить скрипт, як є, і зациклиться.

Альтернативні технології Ajax:

- Java-аплети, пізніше технологія JavaFX;
- технологія Silverlight корпорації Microsoft;
- протокол WebSocket.

Можемо згадати і технологію Flash. Вона використовується в вигляді різноманітних додатків на всіх популярних Інтернет-ресурсах, але поступово на заміну цій технології поступово приходять HTML5.

Стек технологій Flash (Macromedia Flash)це : ActionScript 3, Adobe Flex і Flash Remoting. Вони становлять технологічну основу RIA (Rich Internet



Applications). Цю технологію активно просувала Macromedia , яку поглинено компанією Adobe.

Технологія Flash підходить для самих різних додатків – від комп'ютерних ігор до складних інтерфейсів бізнес-додатків. В межах даної технології реалізовані потужні засоби підтримки графіки, відсутні в базових засобах Ajax (хоча дедалі зростаючу кількість можливостей реалізується в рамках стандарту HTML5 та CSS3).

Отже, ми розглянули технологію Ajax і дізнались, що вона використовується при створенні багатofункціональних сайтів, переваги та недоліки даної технології, прогрес не стоїть на місці і на заміну цим так розповсюдженим зараз технологіям найближчим часом можуть прийти нові такі, як HTML5.

## **2.2. Google Cloud Platform**

Google Cloud– платформа хмарних обчислень від Google, яка пропонує хостинг з опорної інфраструктури, що використовує сам Google для користувачів продуктів, (Google Пошук і YouTube) (рис. 2.2). Ця технологія поєднує в собі всі три категорії SaaS, IaaS, IaaS.

Хмарна платформа являє собою набір модульних хмарних сервісів, які дозволяють створити що-небудь починаючи з простих сайтів і закінчуючи складними додатками. Платформа використовує «будівельні блоки», які сильно спрощують роботу [3].



Рисунок 2.2 - Логотип Google Cloud Platform

Google Cloud Platform є частиною пакету корпоративних рішень від Google. Вона надає набір модулів на хмарній технології, що включають

інструменти для розробки, хостинг і обчислювальну систему, хмарні системи зберігання даних, інтерфейси для перекладу та прогнозування (рис. 2.3).

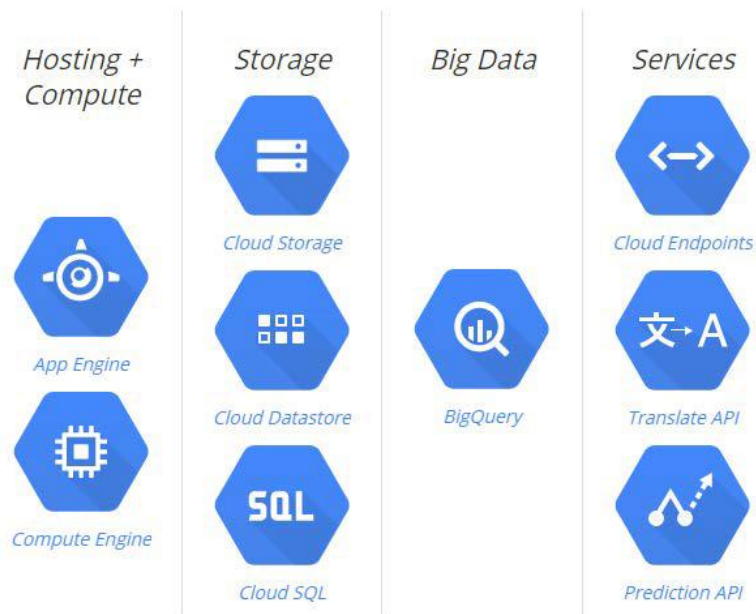


Рисунок 2.3 - Найпопулярніші сервіси в Google Cloud Platform

Для розробників ігор Google Cloud Platform дозволяє легко побудувати масштабну гру, не турбуючись про хостинг, обчислювальну систему, бази даних, тощо (рис. 2.4)

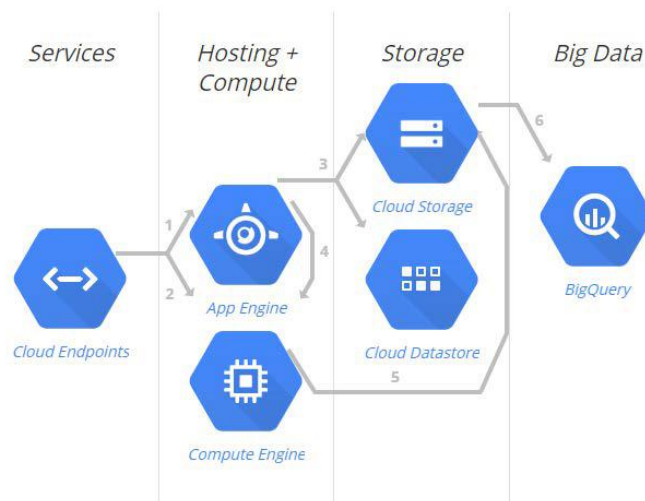


Рисунок 2.4- Схема інфраструктури, яку використовує гра на Google

## **.Compute Engine**

Google Compute Engine – ще один сервіс, обчислювальна система, яка дозволяє створювати великі обчислювальні кластери, які складаються з крос-машини з великою пропускнуою здатністю. Вони з'єднуються між собою та можуть досягати потужності в тисячі ядер. Користувач має можливість підключати обчислювальну систему до інших послуг Google, а також інші сервіси, які не пов'язані з Google. Для таких дій потрібно використовувати глобальну приватну оптоволоконну мережу Google.

Користувачі даного сервісу можуть випробувати швидкість роботи і подивитись на обчислювальну систему з середини, так як вихідний код Apache Hadoop, що використовується на віртуальних машинах Google Compute Engine є відкритим (рис. 2.5).

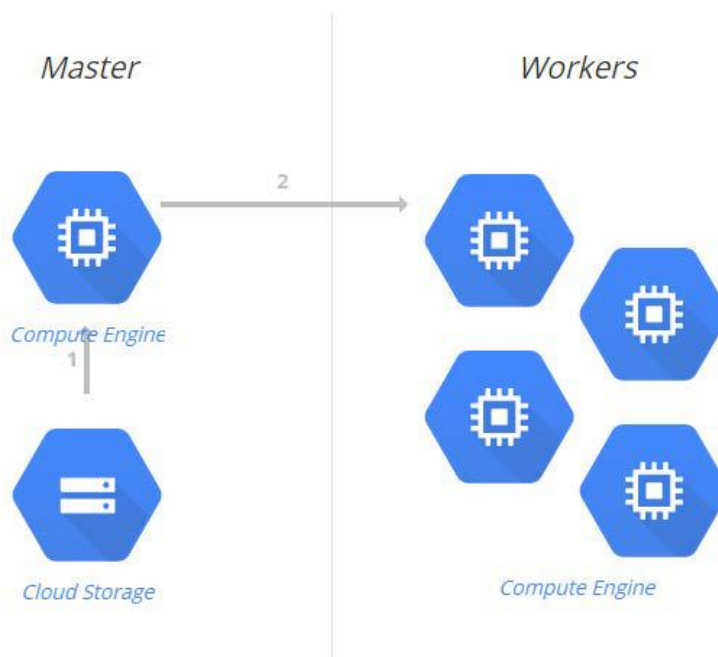


Рисунок 2.5- Схема інфраструктури Google Compute Engine

## Google App Engine

Використовуючи платформи Google Cloud, ми отримуємо:

- інтуїтивно зрозумілий інтерфейс для керування створенням додатку;
- готову інфраструктуру: хостинг, движок і база даних (рис. 2.6).

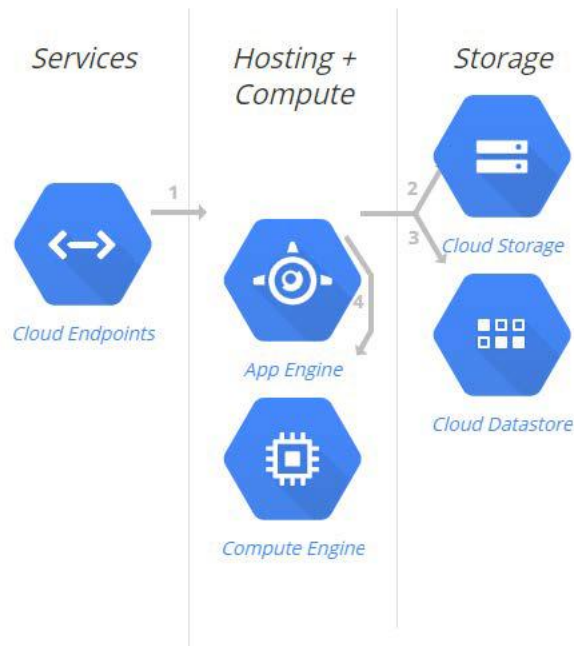


Рисунок 2.6.- Схема інфраструктури Google App Engine

Під словами управляється автоматично розуміється те, що Google потурбується про адміністрування баз даних, конфігурації сервера, сегментування і балансування навантаження, що дозволяє приділяти більше уваги розробці.

- Можливість вибрати варіант зберігання даних: традиційні бази даних з MySQL, використовуючи Cloud SQL, NoSQL сховища даних, або зберігання об'єктів, використовуючи Cloud Storage.
- App Engine надає вам більшу продуктивність роботи шляхом усунення необхідності записувати шаблонний код. Сервіси для керування, такі як Task Queues, Memcache і Users API, дозволяють побудувати будь-які додатки, приділяючи більшу частину уваги на основний код програми.
- Підтримка популярних мов програмування, ви можете використовувати: Python, Java, PHP і Go. Можливість використання вже існуючих структур, таких як Django, Flask, Spring і webapp2. Можливість локальної розробки використовуючи конкретну мову SDK. Отримуєте можливість з'єднати ваш власний додаток з Compute Engine та інтегрувати інші знайомі технології, такі як Node.js, C ++, Scala, Hadoop, MongoDB, Redis і та ін.

- Можливість використання інструментів, які ви знаєте, у тому числі Eclipse, IntelliJ, Maven, Git, Jenkins, PyCharm та ін. App Engine SDK дозволяє тестувати програми локально в модельованому середовищі, а потім розгорнути програму за допомогою простих інструментів командного рядка або засобами робочого столу.

Важливо додати, що додатки створені за допомогою Google App Engine мають гарну оптимізацію, а це означає що з ними зручно працювати як з ПК, так і з планшета або смартфона.

В порівнянні з простою технологією Ajax ми отримуємо однозначну відповідь, платформа Google Cloud значно зручніший інструмент при створенні веб-додатка і виграє у всіх категоріях оцінювання. Єдине «але» це те, що за такий набір функцій потрібно платити, але все ж таки затрачені кошти варті тих можливостей, які надає Google.

### 2.3. Microsoft Azure: Cloud Computing Platform & Services

Microsoft Azure (Windows Azure до 25 березня 2014 р.) – хмарна платформа системи обчислень та інфраструктури, розроблена Microsoft, для створення, розгортання та управління додатками і послугами через глобальну мережу Microsoft. Це можуть бути як PaaS так і IaaS. Вона підтримує безліч різних мов програмування, інструментів і механізмів, у тому числі і Microsoft, так і програмне забезпечення інших виробників і систем [4].

Розглянемо найпопулярніші сервіси, які нам пропонує платформа Microsoft Azure (рис. 2.7)



Рисунок. 2.7.- Логотип Microsoft Azure

## Популярні сервіси Azure.

*Microsoft Azure Virtual Machines* – сервіс, який дозволяє розгортати широкий спектр обчислювальних рішень. Якщо вам потрібно працювати з великими обсягами даних, то можна запуснути віртуальну машину, використовуючи, наприклад: Windows, Linux, SQL Server, Oracle, IBM, SAP, і BizTalk.

### *Створення додатків SaaS для підтримки бази даних.*

Програмне забезпечення для розробників додатків, як послуга (SaaS) в платформі Microsoft Azure, надає можливість використовувати SQL бази даних для робочих навантажень з непередбачуваним споживанням ресурсів. Для цього існує еластична модель. Ця база надає можливість об'єднати ресурси в одній групі баз даних на відміну від сталої моделі.

*Machine Learning Azure* – сервіс дозволяє вирішувати поставлені проблеми, використовуючи лише браузер. Сервіс дає змогу за допомогою простого перетягування в робочу область блоків з даними або алгоритмами і вирішення популярних задач, використовувати стандартні шаблони або створити свої за допомогою мовних пакетів ( Ruby і Python). ,

### *Web Apps в Microsoft Azure*

Сервіс поділяється на 3 категорії:

- Web Apps (веб-додатки).
- Mobile Apps (додатки, які спеціалізовані на мобільні пристрої).
- RemoteApp (додатки, які запускаються по технології віддаленого робочого столу).

Оскільки нам цікаве створення саме веб-додатку, далі мова піде саме про цю категорію на даній платформі. Отже, якщо ми хочемо створити веб-додаток, ми отримуємо:

- Інтуїтивно зрозумілий інтерфейс (конструктор) для подальшого керування створенням додатку.
- Можливість швидко створювати, розгортати і керувати додатком з використанням єдиного інтерфейсу. Використовувати існуючі навички в

програмуванні засобами – .NET, Java, PHP, NodeJS або Python. Прискорити розробку з доступом до багатої галереї API, конекторів і логіки, доступної в Azure Marketplace.

- Підключення до будь-якої служби і розблокування даних. Підключення веб-додатку для корпоративних систем або SaaS за лічені хвилини з вбудованими конекторами, вибравши один з більш ніж 50 конекторів для корпоративних систем, таких як SAP, Siebel, Oracle і в популярних корпоративних SaaS послугах, таких як Salesforce і Office 365 в популярних Інтернет-сервісах, таких як Facebook, Twitter і Dropbox.

- Проста інтеграція даних між хмарними і автоматизованими бізнес-процесами за лічені хвилини за допомогою простого візуального проектування. Легке інтегрування логічного додатку з будь-якого веб-додатку через стандартний інтерфейс REST. Створення складних інтегрованих корпоративних додатків, B2B рішення, використовуючи EDI, правила движка, і більше, залежно від галузі, BizTalk платформи.

- Неперевершена продуктивність праці розробників. Оптимізована для DevOps, з постійною підтримкою інтеграції для Visual Studio онлайн і Github, так що ви можете зосередитися на поліпшенні швидкодії вашої програми, не хвилюючись про інфраструктуру. Впровадження оновлення в додаток, перед цим протестувавши його працездатність, є можливість відкату.

- Найвищий рівень безпеки ваших даних, можливість робити резервні копії, легке адміністрування і 25 річний досвід Microsoft в обслуговуванні.

Прямим конкурентом цієї платформи є Google Cloud Platform. Обидві платформи мають багато однакових функцій, дуже перспективні, з гігантськими можливостями по створенню веб-додатків. Вони відрізняються реалізацією користувацького інтерфейсу і деякими сервісами, але надають майже однакові можливості в створенні веб-додатків.

## Висновки до другого розділу

В цьому розділі проведено аналіз можливостей та властивостей популярних системи та платформи, які використовуються для створення та роботи з веб-додатками

Розглянули технологію Ajax, яка є базовою в створенні веб-додатків та багатофункціональних сайтів; показали переваги та недоліки даної технології. Далі розглянули системи та платформи – Google Cloud Platform, Microsoft Azure. Було розглянуто найпопулярніші сервіси, які нам пропонує платформа Microsoft Azure, що є хмарною платформою системи обчислень та інфраструктури, розробленою Microsoft.

Підсумовуючи, можна сказати, що створювати свій вебдодаток набагато зручніше, використовуючи платформи, які були представлені в цьому розділі. Вони надають величезний спектр сервісів для створення вашого додатку. Важливим в них є інфраструктура, яка включена в ці сервіси, що дозволяють економити дуже багато часу та ресурсів, постійно розвиватися та надає допомогу в вирішенні питань. Хоча за такі переваги і доведеться заплатити, але однозначний висновок, що такі сервіси варті своїх грошей і в повному обсязі виправдовують вкладені затрати.



## **РОЗДІЛ 3. РОЗРОБКА WEB- ДОДАТКУ.**

### **3.1. Огляд систем цифралізації гуртожитків**

Відомо, що вебдодаток є комп'ютерною програмою, яка виконує певні функції, використовуючи веб-браузер у якості свого клієнта. Додаток може бути простим (контактна форма на веб-сайт), або складним (багатокористувацький мобільний ігровий додаток).

Програмний продукт, який було розроблено в цій кваліфікаційній роботі є вебдодатком для збору та обробки даних, щодо поселення студентів та обліку поточного стану завантаженості кімнат гуртожитку в режимі реально часу на основі хмарних обчислень.

Призначення додатку:

- здійснення процесу е-поселення студентів;
- системи ідентифікації студентів, які проживають у гуртожитку;
- зменшити кількість паперової документації;
- зробити процес поселення у гуртожитки більш прозорішим;
- прискорить процес з метою своєчасного прийняття адміністративних та управлінських рішень.

Система дозволить автоматизувати створення звітів, зведених відомостей та інших документів, передбачених документообігом в роботі деканата.

**Функціональні можливості вебдодатку:**

- 1) зберігання інформації про користувачів (студентів,гуртожитку), можливість налаштування рівня доступу окремо для кожного користувача;
- 2) зберігання, введення, редагування та видалення інформації;
- 3) заповнення Google -форми з автоматичною підгрузкою даних (ограничение на відправку -100 листів на день);
- 4) створення Google Таблиці з інформацією про поселених студентів;
- 5) формування Google Excel у вигляді списку студентів, які заповнили форму;
- 6) формування договору у Google документах.

Аналіз Інтернет - систем гуртожитків, який було проведено, показав, що існує багато систем з різноманітними функціональними можливостями та дизайном, але систем, призначених для електронного поселення з використанням мобільних пристроїв немає. Існуючі системи не дають можливості обрати гуртожиток, кімнату, створити договір на поселення, відслідкувати за допомогою QR-коду кількість студентів та гостей у гуртожитку.

Було розглянуто кілька діючих систем, а саме, це прикарпатського національного університету імені Василя Стефаника, платформу цифралізації гуртожитку Київської Могилянської Академії.[7,8 ]

Розглянути системи містять багато корисної інформації, але мінусами цих систем є те, що працюють вони тільки під управлінням операційної системи Windows. Жодна з розглянутих Internet-систем не задовольняє критеріям надання функціональних можливостей для поселення в гуртожиток.

Модуль електронного поселення та обліку студентів у гуртожитках університету є актуальною задачею, вирішення якої приведе до покращення життя студентів та зменшення навантаження на працівників деканату.

Процеси поселення та обліку студентів на сьогодні не класифіковані і не стандартизовані.

Під час процесу поселення в гуртожитках утворюються черги студентів та їх батьків , що створює велику скупченість людей. Деканати та коменданти гуртожитків в цей період завантажені великим напливом студентів. Модуль електронного поселення та обліку студентів у гуртожитках значно полегшить цей процес, зменшить кількість паперової документації, зробить процес поселення у гуртожитки більш прозорішим для студентів та прискорить цей процес [ 2 ].

Щоб досягнути поставленої мети під час дослідження потрібно вирішити такі задачі:

- проаналізувати нормативно – правову базу, що забезпечує проживання студентів у гуртожитках на загальнодержавному та університетському рівнях;

- виділити способи та засоби досягнення поставленої мети, які будуть адекватними матеріальному рівню університету та освітній системі України;
- проаналізувати процес поселення студентів у гуртожитки, виділити місця, що потребують багато часу та операцій, які можна автоматизувати;
- провести тестування по зручності користування розробленим додатком [3].

Майже всі університети нашої держави пропонують за вигідними цінами проживання в гуртожитках під час навчання. Зазвичай студентів поселяють на 10 місяців – навчальний рік, після чого студент має залишити гуртожиток. Після літніх канікул студенти, що хочуть проживати в гуртожитку, повинні пройти нелегкий бюрократичний процес. Кожного року при поселенні в гуртожитки університету студенти і працівники університету зазнають психологічного і фізичного удару, перевантаження. Плутанина, спричинена складним документообігом, непрозорістю процесу поселення, може заморозити перші дні навчального процесу, які можуть виявитись фатальними у подальшому навчанні, та викликати стрес у працівників університету, що буде вливатись на продуктивність їхньої праці [4].

### **3.2. Огляд Google- продуктів**

Для розробки вебсистеми були використані Google- продукти: Sites, Forms, Drive, Sheets, Apps Script, Docs.

1. *Google Drive*, це сховище даних, яке дозволяє користувачам зберігати свої дані на серверах у хмарі і ділитися ними з іншими користувачами в Інтернеті, включає Google Документи, Таблиці та Презентації, офісний пакет. Для синхронізації файлів між комп'ютером користувача і хмарним сховищем необхідне програмне забезпечення Google Drive ('клієнт') на пристрої користувача..Google Drive доступний в автономному режимі браузера Google Chrome через додаток Chrome.

Додаток Google Drive для Android підтримує офлайн редагування документів і таблиць підтримується через автономні мобільні додатки.

Google Drive включає в себе систему обміну файлами. Файли і теки можуть спільно використовуватись приватно конкретними користувачами, використовуючи свої @gmail.com адреси електронної пошти. Для спільного використання файлів з користувачами, які не мають облікового запису Google потрібно зробити файли «доступними за посиланням»..

Docs, Sheets and Slides являють собою безкоштовний вебофісний пакет, пропонуваний Google і інтегрований у Google Drive. Він дозволяє користувачам створювати і редагувати документи онлайн, співпрацюючи в режимі реального часу з іншими користувачами

2. *Google Документи* — розроблений Google безкоштовний хмарний офісний пакет, що включає текстовий редактор, табличний редактор і службу для створення презентацій.(Вікіпедія)

Доступна велика кількість засобів форматування: зміна розміру і стилю шрифту, вибір кольору та оздоблення, створення списків і таблиць, вставка зображень, посилань і спеціальних знаків.

Зберігаються документи автоматично, по ходу внесення змін, але кожна правка записується, і можна користуватися функцією скасування і повернення змін так само, як і у звичайному текстовому редакторі. Існує можливість завантажувати файли на сервер і скачувати з нього у різних форматах.

3. *Google Форма*. У додатку *Google Диск* є функція «Google Форма», що створена для роботи із формуванням різного роду блоків питань. Дану форму використовують для створення системи звітів, для зменшення витрат часу на опис та здачу паперових звітів.

4. *Google Sheets*. *Google Sheets*.

Для збереження інформації додатку потрібна база даних яка може містити в собі велику кількість інформації, зможе обслуговувати велику кількість користувачів одночасно та буде добре захищеною. Хмарне сховище даних (англ. cloud storage) — модель онлайн-сховища, у якому дані зберігаються на численних розподілених у мережі серверах, які надаються клієнтам, переважно, третьою стороною. Дані зберігаються і обробляються в так званій «хмарі», яка

є, з погляду клієнта, одним великим віртуальним сервером. Фізично такі сервери можуть розташовуватися віддалено один від одного географічно.(Вікіпедія).

Переваги:

- Можливість доступу до даних із будь-якого комп'ютера, який має вихід в Інтернет.
- Можливість спільної роботи з даними.
- Висока ймовірність збереження даних навіть у випадку апаратних збоїв.
- Клієнту не потрібно займатися придбанням, підтримкою та обслуговуванням власної інфраструктури для зберігання даних, що, зрештою, зменшує загальні витрати виробництва.

Створювати прототипи різних моделей для перевірки гіпотез набагато швидше, ніж програмувати те саме на СУБД MySQL. Сервіс має гірші можливості, порівняно з повноцінними електронними таблицями, такими, як Microsoft Excel, але їх цілком достатньо для простих дій з даними.

### 3.3. Мова для програмування в хмарі.

Google Apps Script — скриптова платформа, розроблена в Google для розробки легких вебдодатків на платформі Google Workspace/.Проекти Apps Script запускаються в інфраструктурі Google на стороні сервера. Згідно Google, Apps Script «забезпечує прості шляхи для автоматизації задач на перетині продуктів Google та сторонніх сервісів». Apps Script також являється інструментом для написання розширень для Google Docs, Sheets та Slides.(Вікіпедія)

**Apps Script** дозволяє використовувати для розробки HTML, CSS та JavaScript, не вимагаючи вивчення нового пропрієтарного фреймворку.

Apps Script є єдиною малокодовою платформою, що дозволяє швидко і легко розробляти бізнес-рішення, націлені на інтеграцію з Google Workspace, розширення його функціоналу та автоматизацію завдань. Доступ до Apps Script можливий за допомогою будь-якого облікового запису Gmail.

Основне завдання Google Apps Script полягає в тому, щоб дати вам можливість створювати веб-програми і навіть прості веб-сайти, які будуть зберігатися в хмарі Google.

По суті Google Apps Script дає можливість програмування в хмарі.

Спочатку цей сервіс створювався для взаємодії з іншими сервісами Google (таблиці, документи, пошта (Gmail), презентації, ...).

Для взаємодії з цими сервісами можна або написати якусь веб-програму на нашому хостингу, або ми можемо скористатися Google Apps Script. Завдання цього сервісу, це автоматизація роботи з іншими сервісами Google. У міру розвитку Google Apps Script з'являлися нові можливості і це стало деяким аналогом програмування у хмарі з великою кількістю можливостей, які можна використовувати..

Програмування Google Apps Script відбувається спеціальною мовою google script (gs). Ця мова програмування не що інше, як мова Javascript, тільки з деякими особливостями. Основні функції Javascript Google Script підтримує, але є деякі додаткові можливості, але також є деякі функції, які в google script відсутні. Головне, чому варто використовувати Google Apps Script, для створення веб-застосунків, вам не потрібен свій хостинг і веб-сервер. Це все безплатно. Тобто, прості завдання автоматизації, створення простих веб-сторінок - все це можна робити за допомогою Google Apps Script.

Існують 2 варіанти, як можна працювати скриптами в Google Apps Script.

1). Скрипти, які вбудовані в ту програму, з якою ми працюємо.

Приклад: якщо ми працюємо з гугл таблицями то можна зробити так, щоб вони були вбудовані в цю таблицю.

Для того, щоб це зробити, потрібно перейти в Інструменти та вибрати "Редактор скриптів". Відкриється окремий проект, який буде прив'язаний до цієї програми.

Як окремим файлом на вашому Google диску не існуватимуть.

2). Створення та робота зі скриптами на сайті *script.google.com*.

Скрипти будуть окремим файлом, який зберігатиметься на вашому Google

диску. Сам проєкт Google Apps Script, по суті, простий файл, який зберігається на диску Google.

Ось такі два варіанти роботи з Google Apps Script є.

Google Script базується на програмуванні Javascript. Більшість команд, які є в Javascript, підтримуються Google Script. Але також тут є додаткові команди, які можна використовувати тільки в Google Script.

Одну з таких команд ми зараз і побачимо.

```
function myFunction {  
    Logger.log('111');  
}
```

За допомогою цієї команди можна вивести щось у лог Google Apps Scripts.

Таких функцій можна створити скільки завгодно. Проєкти складаються із таких функцій.

У Google Apps Script запускаємо виконання функцій. Працюючи з Google Apps Script буде виконуватись робота з сервісами.

Сервіси - це деякі програмні класи, які мають якісь способи, якості, з якими можна працювати і щось за допомогою цих сервісів створити.

По суті, все програмування Google Apps Script зводиться до завдання взаємодії з якимись сервісами.

Наприклад з сервісами G Suite. Це послуги, які надає нам Google( Календар, контакти, документи, таблиці, тощо).

З усіма цими сервісами Google ми можемо взаємодіяти за допомогою Google Apps Script. Тут є опис цих класів, методи їх викликів. Сервіс доє змогу взаємодіяти з більш сучасними Google сервісами- Adsense, Analytics, Big Query та ін.

3) Це скриптові послуги. Це JDBC для взаємодії з сторонніми базами даних. HTML – для взаємодії з HTML сторінками, і.т.д.

На лістингу представлені фрагменти кодів етапів поселення :

ЛісИТНГ коду на мові Apps Script.

```
    Const agreementTemplateId =
    "1XpTPkIPK5XqcfLf5Cg4CrbxzBLnO8bDb7BvOVS-eOMY"
    const agreementSaveFolderId = "1-HdUhTILdqi84WzLCz3ohMmGR-
    IK2UC9"

    const controlSpreadsheetId =
    "1OVv3Y6GjAuRP4u3eBeq3vNU7x_YPHQVTjIUtlVhmo9E"
    const roomsSheetName = "Кімнати"
    const spreadsheetDataStartRow = 3
    const manRoomAvailabilityColumn = 1
    const manRoomNumberColumn = 3
    const womanRoomAvailabilityColumn = 5
    const womanRoomNumberColumn = 7

    const agreementFormId =
    "1sVjCbi7VexvGHOqdYcXUkspIxWY3O_QAWe6rpHwgcQo"
    const manRoomsSelectId = 282071382
    const womanRoomsSelectId = 1168871159

    const normalizeDate = (dateString) => {
    date = new Date(dateString)
    return `${date.getDate()}.${date.getMonth()+1}.${date.getFullYear()}`
    }

    const generateAgreementDocument = (formValues) => {
    const useIdCard = formValues[5] == "ID картка"
    const notAdultStudent = formValues[14] == "Так"

    const student_name = formValues[2]

    const id_card_code = formValues[11]
    const id_card_issued_at = normalizeDate(formValues[13])
```



```

const id_card_issuer = formValues[12]

const passport_code = formValues[7]
const passport_number = formValues[8]
const passport_issued_by = formValues[9]
const passport_issued_date = normalizeDate(formValues[10])

const legal_representative = formValues[15]
const representative_passport_code = formValues[16]
const representative_passport_number = formValues[17]
const representative_passport_issued_by = formValues[18]
const representative_passport_issued_date =
normalizeDate(formValues[19])

const student_legal_document = useIdCard ?
`ID картка: ${id_card_code}, видана ${id_card_issued_at}, органом:
${id_card_issuer}` :
`Паспорт: ${passport_code}${passport_number}, виданий
${passport_issued_by}, ${passport_issued_date}`

const representative_agreement = notAdultStudent ?
`Я, ${legal_representative}
паспорт: серія ${representative_passport_code} №
${representative_passport_number}, ким і коли виданий
${representative_passport_issued_by}, ${representative_passport_issued_date}.
Даю згоду на укладання даного договору з моїм сином (моєю
донькою) ${student_name}` :
"

const columnsMapping = {
  "agreement_date": normalizeDate(formValues[20]),
  "student_name": student_name,
  "legal_representative": legal_representative,

```

```

    "agreement_at": normalizeDate(formValues[21]),
    "agreement_to": normalizeDate(formValues[22]),
    "student_address": formValues[3],
    "student_phone": formValues[4],
    "student_legal_document": student_legal_document,
    "tax_number": formValues[6],
    "student_email": formValues[1],
    "representative_passport_code": representative_passport_code,
    "representative_passport_number": representative_passport_number,
    "representative_passport_issued_by":
representative_passport_issued_by,
    "representative_passport_issued_date":
representative_passport_issued_date,
    "representative_agreement": representative_agreement,
  }
  const createdAt = normalizeDate(formValues[0])
  const templateFile = DriveApp.getFileById(agreementTemplateId)
  const saveFolder = DriveApp.getFolderById(agreementSaveFolderId)

  const agreementFile =
templateFile.makeCopy(`${columnsMapping.student_name}-${createdAt}`,
saveFolder)
  let agreementDocument =
DocumentApp.openById(agreementFile.getId())
  let agreementBody = agreementDocument.getBody()
  Object.keys(columnsMapping).forEach((key) =>
agreementBody.replaceText(`{ ${key} }`, columnsMapping[key]))

  agreementDocument.saveAndClose()

  return agreementDocument
}

```

### 3.4. Google Таблиці, як база даних.

Для збереження інформації додатку потрібна база даних .

Додаток Таблиці Google для роботи з електронними таблицями, входить до складу безкоштовного вебпрограмного пакету, пропонованого компанією Google у межах служби Google Диск. Інтерфейс додатка схожий з інтерфейсом Microsoft Excel, що входить до складу Microsoft Office та сумісний з форматами файлів Microsoft Excel [Вікіпедія].

Google Sheets часто використовують як редактор таблиць. У той же час їх можна використовувати як найпростішу базу даних.

Оскільки Google Таблиці доступні через Google Диск, спочатку потрібно увійти в систему за допомогою облікового запису Google, щоб створювати, редагувати, зберігати та ділитися файлами. Обліковий запис Google є уніфікованою системою входу, яка надає доступ до каталогу продуктів Google а Gmail не потрібен для використання Google Диска та Таблиць. Будь-яка електронна адреса пов'язана з обліковим записом Google.

Google Sheets пропонує основні функції, які можна очікувати при роботі з електронними таблицями:

- Налаштування електронної таблиці та даних (з можливістю автоматичного заповнення).
- Робота з рядками, стовпцями та клітинками.
- Впровадити функції, макроси та сценарії для складних обчислень.
- Додайте діаграми / графіки, зведені таблиці та зображення.
- Імпорт та / або пошук даних у електронних таблицях [6].

Переваги використання Google Таблиць:

- Працюємо тільки з одним документом, навіть із декількох пристроїв, (платформ або місцеположень), тому що файли зберігаються в хмарі (Google Drive). Зміни зберігаються автоматично, а також доступне

редагування в автономному режимі (за допомогою мобільного додатка та веб-браузера Google Chrome).

- Ділімся файлами з іншими для спільної роботи, редагування в реальному часі, коментування та спілкування в чаті. Історія версій Google Sheets (яка вбудована) відстежує всі зміни внесені людьми та надає користувачам можливість відновити файл до попередньої версії.
- Інтеграція до Google Forms . Дає змогу створення або вставки опитувань, зворотного зв'язку, анкет опитувань у презентаціях електронних таблиць; Google Translate (функції копіювання для перекладу мов) або Google Finance (автоматично знаходить і вводить вказану фінансову інформацію)
- Безкоштовна для індивідуального використання.

Google Sheets мають певні переваги, але це не справжня заміна Microsoft Excel, яка включає:

- Додаткові параметри для шаблонів, налаштування та вдосконалених інструментів редагування / форматування.
- Автоматичне налаштування формул при додаванні або видаленні категорій.
- Управління та обробка величезних обсягів даних.
- Широкий вибір діаграм та графіків для подання інформації.
- Розширені функції та формули, ідеально підходять для фінансів, статистики, науки, техніки тощо.

### **3.5. Розробка діаграми послідовності**

Діаграма послідовності – різновид діаграми в UML. Діаграма послідовності відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень. На рисунку 3.1 представлена діаграма послідовності реєстрації.

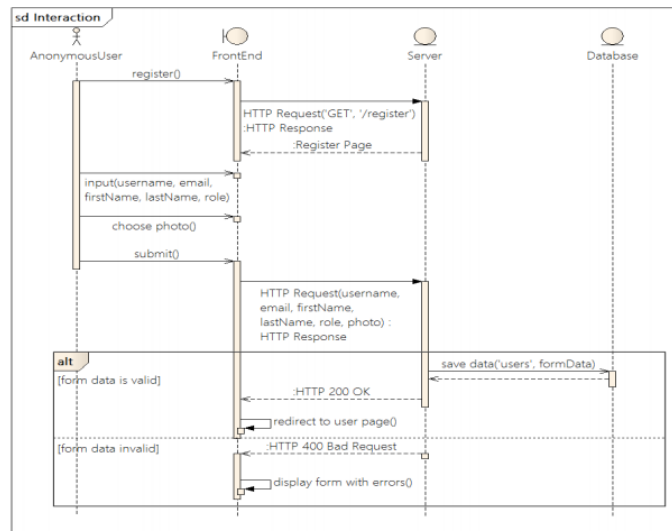


Рисунок 3.1 – Діаграма послідовності реєстрації

Користувач переходить за посиланням на сторінку реєстрації, після чого браузер відправляє запит на сервер, а сервер повертає сторінку. Користувач заповнює поля, які перенесуть інформацію з них до бази даних після реєстрації. Після цього браузер передає дані запитом на сервер. Якщо дані вірні, сервер звертається до бази даних та записує дані в неї, якщо дані не вірні, виводить помилку про некоректність введення даних або про поля які не були заповнені. На рисунку 3.2 представлена діаграма послідовності авторизації.

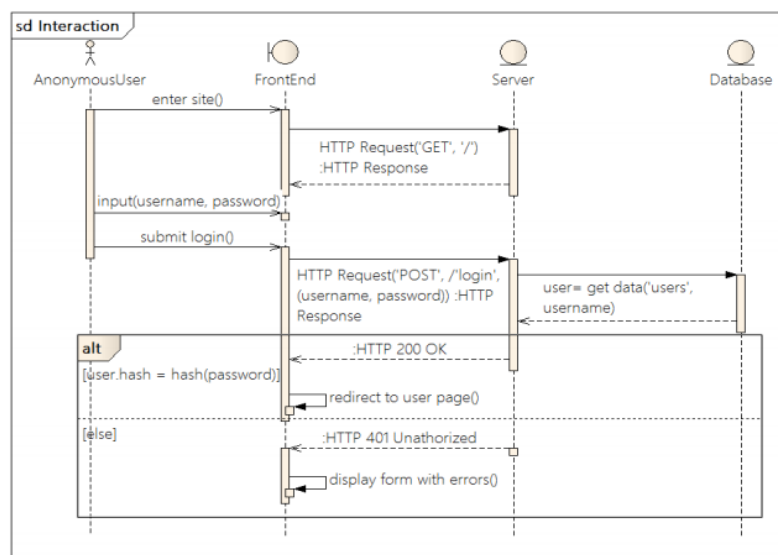


Рисунок 3.2 – Діаграма послідовності авторизації

Користувач заповнює дані, після цього браузер відправляє дані на

сервер, який перевіряє їх з базою даних. Якщо дані вірні сервер передає сторінку користувача, інакше сторінку з помилкою.

### **3.6. Алгоритм роботи системи та результати тестування**

Модуль «Електронне поселення» включає розробку GoogleForm анкети студента, що поселяється в гуртожиток та створення електронного «Типового договору», за даними, які підтягуються з анкети. Для керівництва паралельно створюється Excel сторінка з відповідними даними.

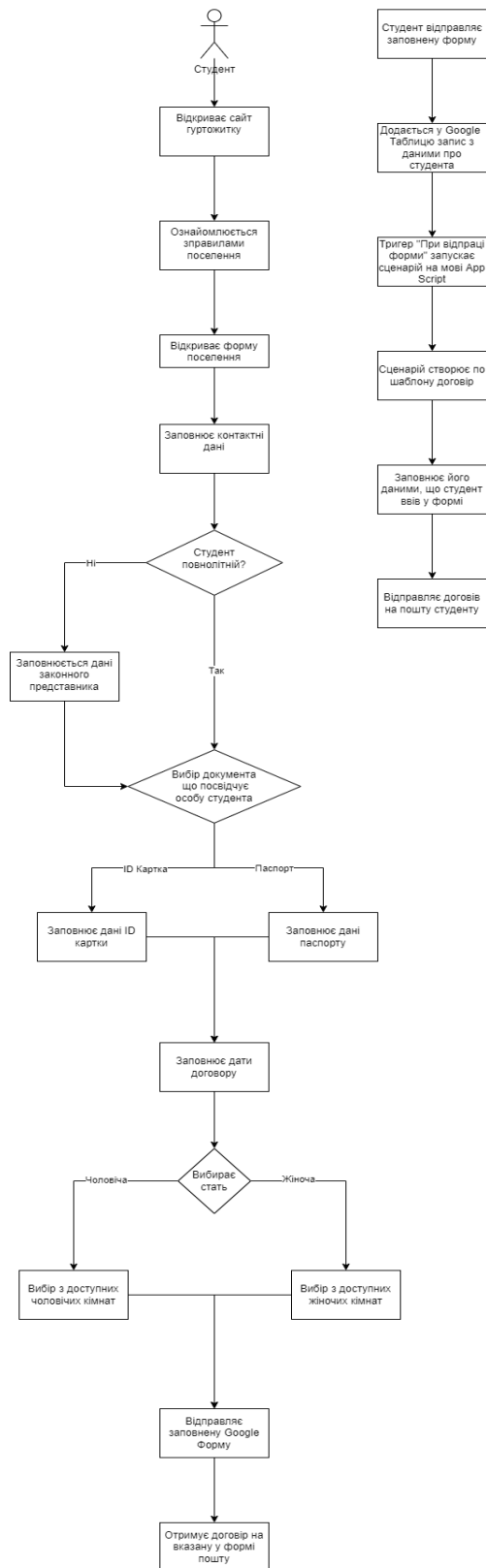
Кожен студент має змогу при заповненні анкети:

- Вибрати кімнату;
- Оформити документ і передати його в деканат в дистанційному режимі;
- Уникнути черги на затвердження документів в деканаті та прийом у коменданта для вирішення особистих питань.

Задача електронного поселення, включає також розробку QR-коду студента, що проживає у гуртожитку і буде розміщено на договорі.

Алгоритм роботи системи має наступні етапи:

1. Щоб подати електронну заяву на поселення до гуртожиток, студент на сайті (рис.3.4) може ознайомитись з інформацією про гуртожиток та переглянути правила проживання та користування гуртожитками. Це він зможе зробити за допомогою QR-коду, при скануванні QR-коду він буде переспрямований на Google Site, або зайти на сайт за посиланням на головній сторінці сайту.



3. 3- Блок –схема етапів поселення

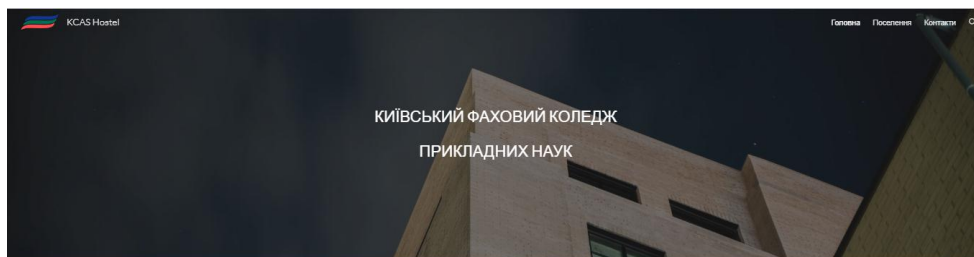


Рисунок 3.4.- Головна сторінка сайту

2. На сторінці "Поселення" студент має змогу ознайомитись з умовами проживання, фотографіями кімнат, що можна та не можна брати з собою в гуртожиток, подати заявку на поселення та сплатити за проживання онлайн (, рис.3. 5) .

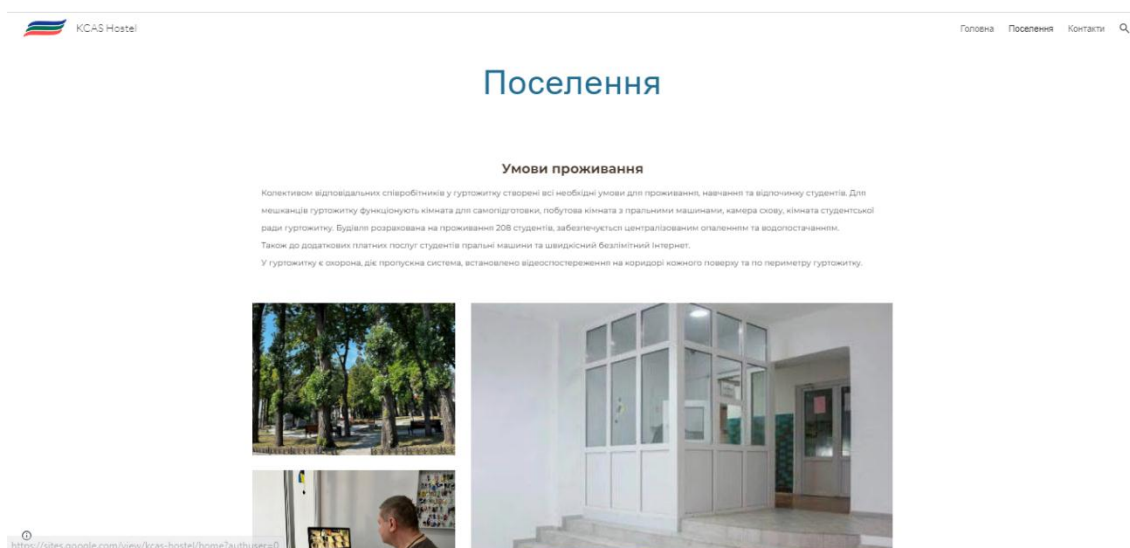


Рисунок 3.5 - Інформаційна сторінка «Поселення»



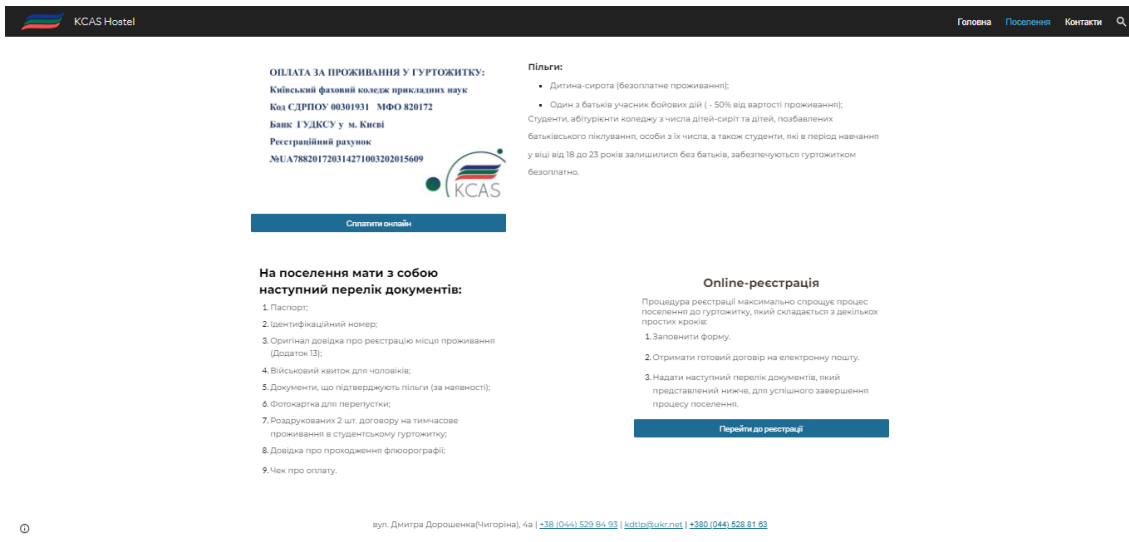


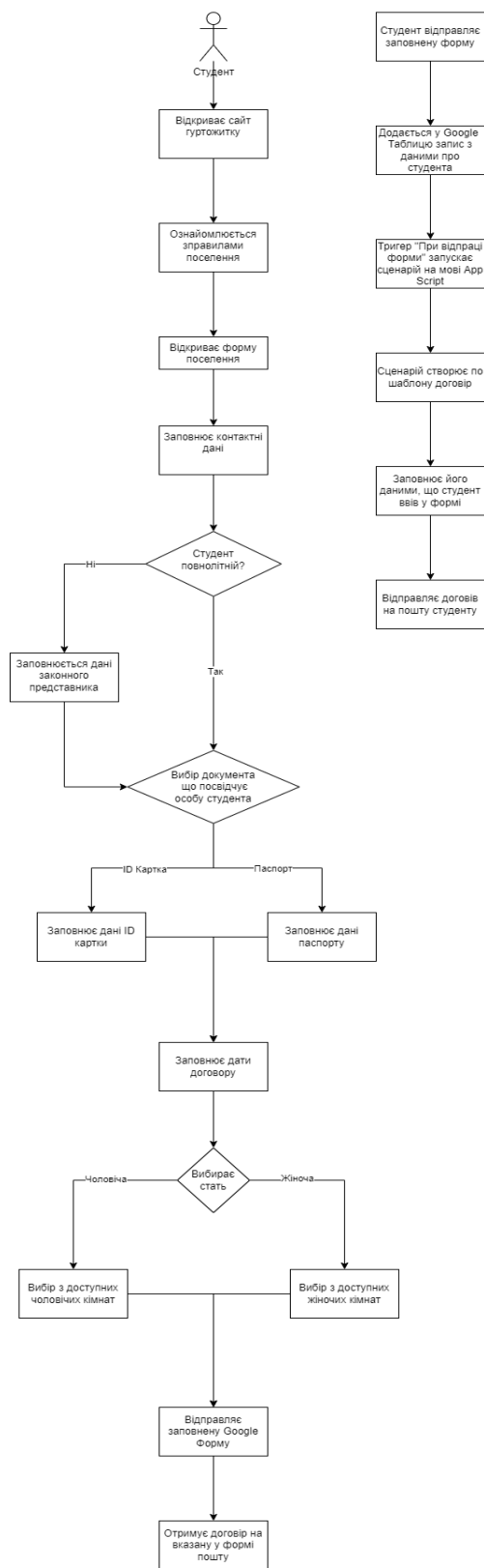
Рисунок 3.6- Google Site для навігації студента.

3. Натиснувши на кнопку "Перейти до реєстрації", студент буде перенаправлений на Google Form, де студенту потрібно ввести свої дані (рис. 3.7)

The screenshot shows a Google Form titled 'Заповнення договору' (Agreement completion). The form is for a 'Договір найму жилого приміщення у гуртожитку Київського Фахового Коледжу Прикладних Наук' (Lease agreement for a hostel room at the Kyiv Specialized College of Applied Sciences). The form includes the following fields:

- A text field containing the email address 'r73255621@gmail.com' with a 'Switch account' link and a cloud icon.
- A red asterisk indicating a required question: '\* Indicates required question'.
- An 'Email \*' field with the placeholder text 'Your email'.
- A 'Прізвище, ім'я та по батькові здобувача освіти \*' (Surname, first name, and patronymic of the student) field with the example 'Наприклад: Іванов Іван Іванович' and the placeholder text 'Your answer'.
- A 'Поштова адреса \*' (Postal address) field with the placeholder text 'Наприклад:'.

Рисунок 3.6. Google Form для заповнення даних.



3. 7- Блок –схема етапів поселення

#### 4. Після заповнення даних та відправки форми, відбудеться 3 події:

- Всі дані студента відправляються до Google Таблиці, яку використовуватиме деканат, для створення шахматки заселених студентів (рис.3.8).

літкoмісце	Найім	ПІБ	Стать	ЗВО/фак/інститут	курс, група	№дoгoвoру	кoнтaктний тел	ЗЯЯВА (план)	заселено	Всього місць	Примітки
1										1	
2		ПОМОШКО МАКСІМ ДМИТРОВИЧ	Ч	БА-17			s		1	1	
3		ХРИЩУК НАЗАР ВІТАЛІЙОВИЧ	Ч	БА-17			s		1	1	
4	14	СЕМЕНОВ СТАНІСЛАВ ОЛЕКСАНДРОВИЧ	Ч	БА-17			s		1	1	
5		ЖИГЛІЙ МИКОЛА ВАДИМОВИЧ	Ч	БЕМ-16	ННПТ	46/91-19			1	1	
6		Мартинюк Андрій Олександрович	Ч	БФР-19	ФПП	32/902/19	665620267		1	1	
7			Ч	БІТ-16			s		1	1	
8	15	КУДРЕВИЧ НАЗАР ВОЛОДИМИРОВИЧ	Ч	БА-17			s		1	1	
9		АЛЕКСЄЄВЕЦЬ АНТОН СЕРГІЙОВИЧ	ч	мгМ-20			s		1	1	
10		ВІТЮК НАЗАР ПЕТРОВИЧ	Ч	БПМ-17			s		1	1	
11		ПРЯДКО ІГОР ОЛЕКСАНДРОВИЧ	Ч	БПМ-17			s		1	1	
12	16	ДОМАШЕНКО ВЯЧЕСЛАВ ВЯЧЕСЛАВОВ	Ч	БПМ-17			s		1	1	

Рисунок 3.8 - Google Excel у вигляді документа заселених студентів.

- Створюється договір у Google документах.( рис3.9 ).

Договір № Ст / -  
на найм жилого приміщення у гуртожитку КНУТД на 2021/2022 навчальний рік

м. Київ 26 вересня 2021 р

Київський національний університет технологій та дизайну (далі — Наймодальць та/або КНУТД), в особі заступника проректора з науково-педагогічної діяльності (освітня діяльність) Бугаса В.В., який діє на підставі наказу КНУТД від 27.01.2020 року №12 з однієї сторони та Ляховська Дар'я Дмитрівна, група БПМ\_20 курс 4 (далі — Найміч) з другої сторони, разом – Сторони, а кожна окремо – Сторона, уклали цей Договір найму жилого приміщення у гуртожитку КНУТД на „\_ навчальний рік (далі - Договір) про наступне:

- 1. Предмет Договору**
  - 1.1. Наймодальць надає, Найміч приймає в строкове платне користування жиле приміщення, що являє собою ліжко-місце в кімнаті № 20 у гуртожитку КНУТД №3, яке розташоване за адресою: вул. М.Войцюка, 34 та зобов'язується провести оплату за проживання у гуртожитку у розріді та порядку, передбаченому цим Договором.
- 2. Обов'язки та права Наймодальця:**
  - 2.1. Встановити порядок надати Наймічу ліжко-місце в гуртожитку на період дії Договору;
  - 2.1.2. Забезпечити умови проживання в гуртожитку та твердий і м'який інвентар згідно переліку — відповідно до діючих норм. Забезпечити водо-, електро-, теплопостачання; за умови, що гуртожиток газифіковано - газопостачання;
  - 2.1.3. При пошкодженні і аваріях негайно викликати заходи і ліквідації їх наслідки, а у випадках, якщо пошкодження виникло з вини Найміча чи осіб, які разом з ним проживають або запрошені ними - за рахунок Найміча;
  - 2.1.4. Провести інструктаж з дотримання правил техніки безпеки, винюг пожежної безпеки та санітарних норм, порядку відшкодування матеріальних збитків та сплати за додатково надані комунальні послуги;
  - 2.1.5. Надати ордер на зайняття жилого площі та перепустку для входу до гуртожитку;
  - 2.1.6. **2.1. Наймодальць має право:**
    - 2.1.1. Виконувати вимоги Наймодальця умов цього Договору, Правил внутрішнього розпорядку в гуртожитках КНУТД та Положення про користування гуртожитками КНУТД;
    - 2.1.2. Достроково припинити (розірвати) Договір в односторонньому порядку у випадку порушення Наймічем умов Договору, Правил внутрішнього розпорядку в гуртожитках КНУТД та Положення про користування гуртожитками КНУТД, норм чинного законодавства України, які визначають обов'язки Найміча;
    - 2.1.3. Займовувати вартість за проживання протягом дії Договору та надавати інші додаткові платні послуги;
    - 2.1.4. Складати акт про закінчення Наймічем матеріальний збиток майну чи інвентарю гуртожитку та отримати відшкодування збитку згідно з чинним законодавством України;
    - 2.1.5. Відмовити в поселенні на наступний навчальний рік особам, які порушували умови цього Договору, Правил внутрішнього розпорядку в гуртожитках КНУТД та Положення про користування гуртожитками КНУТД;
    - 2.1.6. Вилучати додаткове електричне обладнання та інші прилади Найміча, що використовуються без відповідного дозволу, до камери скопу;
- 3. Обов'язки та права Найміча гуртожитку:**
  - 3.1. **Найміч зобов'язаний:**
    - 3.1.1. Неухильно дотримуватися вимог Положення про користування гуртожитками КНУТД та виконувати Правила внутрішнього розпорядку в гуртожитку КНУТД;
    - 3.1.2. Прийняти в користування твердий і м'який інвентар згідно переліку;
    - 3.1.3. Користуватися майном, підсобними приміщеннями, місцями загального користування виключно за принципом призначенням на рівних правах з іншими мешканцями;
    - 3.1.4. Сплатувати за проживання. Оплата здійснюється авансово за навчальний рік в сумі 6100 гривень, без ПДВ;
    - 3.1.5. При поселенні надати наказ/акт гуртожитку;
    - 3.1.5.1. медичну довідку для поселення в гуртожиток з Київської міської студентської поліклініки або сімейного лікаря Найміча, томо;
    - 3.1.5.2. квитанцію про сплату за проживання;
    - 3.1.5.3. ордер на зайняття жилого площі;
    - 3.1.5.4. перепустку для входу до гуртожитку;
    - 3.1.6. Протягом 10 днів після поселення надати документи на реєстрацію у паспортний стіл та військовозобов'язаним стати на військовий облік;
    - 3.1.7. Протягом 3-х днів після поселення, в обов'язковому порядку, забезпечити наявність дубляжів ключів від кімнати, (базу) у завідувача кімнатною кімнатою та в разі відсутності це зробити власноруч протягом 10 днів після поселення;

Рисунок 3.9 . – Google форма договору

Договір надсилається на пошту, вказану студентом.

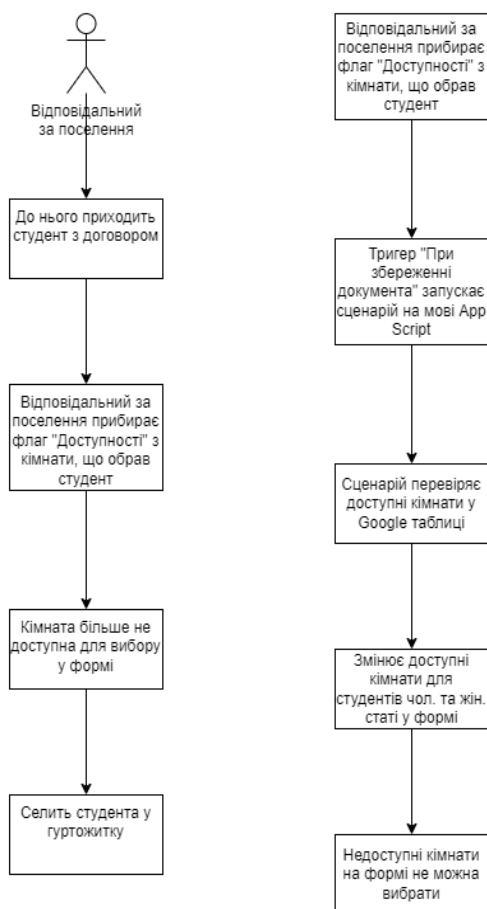


Рисунок 3.10 – Блок-схема роботи «Відповідального за поселення».

Далі студенту залишається роздрукувати та підписати договір.

Все це перенесе деякі процеси, які існують у гуртожитку та в деканаті, у мобільний девайс чи персональний комп'ютер, дозволить ідентифікувати наявність студентів у гуртожитку та зробить сучасною систему входу до гуртожитку та поселення студентів.

## ВИСНОВКИ

В кваліфікаційній роботі було розглянуто питання цифралізації в закладах освіти .

Гуртожиток, як частина вищого навчального закладу , також потребує інновацій. Наша розробка допоможе вирішити одну із задач використання цифрових та web-технологій у сфері освіти, а саме в гуртожитку

В роботі було проаналізовано існуючі інформаційні Інтернет-системи гуртожитків , нормативно – правова база процесу поселення студентів; виявлені місця, що потребують багато часу для вирішення проблем; проведено тестування зручності користування розробленим програмним додатком.

Методами дослідження розробки було обрано ідею використання хмарних технологій з використанням Google-продуктів.

Були розроблені блок-схеми основних етапів поселення студентів у гуртожиток та програмна реалізація цих блок-схем є використанням мови Google Apps Script.

Розроблений нами програмний продукт може бути використаний у будь-якому університеті чи іншій організації, яка потребує автоматизації процесу поселення людей у гуртожитки та обліку поточного стану завантаженості в режимі реально часу. Використання даного програмного продукту приведе до зниження часових витрат персоналу, що займається організацією процесу поселення, а як наслідок зменшення матеріальних витрат на поселення. Електронне поселення – це частина задачі діджиталізацію гуртожитку, яка вже нами реалізована та проходив етап тестування на прикладі гуртожитку №3 КНУТД та Київського фахового коледжу прикладних наук .

Вся проблематика, яка існує у гуртожитку не є безпідставною, вона тягнеться роками і саме створення нашого додатку може стати переломним моментом у відношенні до організації студентського життя.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Моргулець О.Б., Астісова Т. І., Сучасні комп'ютерні технології в організації освітньої діяльності навчального закладу // «Вісник КНУТД», №5 (101) 2016 р, спецвипуск, с.351-359
2. Астісова Т. І. Seo – оптимізація в системі моніторингу web – ресурсів// Т.І. Астісова // Технології та інжиніринг («Вісник КНУТД. Серія Технічні науки») Київський національний університет технологій та дизайну, Україна No1(12), 2023 р. С. 9-17. Scopus Author ID: 6506601603.  
<https://vistnuk.knutd.edu.ua/wp-content/uploads/sites/2/2023/05/1-1-2023.pdf>
3. Методи представлення, збереження та аналізу даних інформаційних систем: колективна монографія / В. Ю. Щербань, С. М. Краснитський, Т. І. Астісова, В. М. Яхно. – Київ : ТОВ "Фастбінд Україна", 2023  
<https://er.knutd.edu.ua/handle/123456789/24076>
4. Астісова Т.І. Аналіз програм та сервісів для поширення сайтів в інтернет-просторі / Т. І. Астісова / Інформаційні технології в науці, виробництві та підприємстві: зб. наук. праць молодих вчених, аспірантів, магістрів кафедри інформаційних технологій проектування. К. : Освіта України, 2020. С.170-175.  
<https://er.knutd.edu.ua/handle/123456789/236641>
5. Астісова Т. І., Глембоцький В.С., Програмне забезпечення для системи ідентифікації студентів/ Т.І. Астісова, В.С. Глембоцький / Інформаційні технології в науці, виробництві та підприємстві: зб. наук. праць молодих вчених, аспірантів, магістрів кафедри комп'ютерних наук та технологій. К. : Освіта України, 2021. С. 208-211 .<https://er.knutd.edu.ua/handle/123456789/236881>
6. Астісова Т. І., Егоров Д.С. Розробка QR-коду для веб додатку «Система ідентифікації студентів»/ Т.І. Астісова, Д.С. Егоров / Інформаційні технології в науці, виробництві та підприємстві: зб. наук. праць молодих вчених, аспірантів, магістрів кафедри комп'ютерних наук та технологій. К. : Освіта України, 2021. С. 214-217.<https://er.knutd.edu.ua/handle/123456789/23652>

7. Астістова Т.І Розробка системи електронного поселення /Т.І. Астістова, Д.Д. Ляховська // Тези V Міжнародної науково-практичної конференції «Мехатронні системи: інновації та інжиніринг – «MSIE-2021» К. КНУТД , 4 листопада 2021р. - С.152
8. AstistovaTetyana, Development of software for accounting for the presence of student attendance// Т.І. Astistova, D. D Liakhovska D. D., М. V. Нукутуук, N.V.Smorzhevsky // Збірник наукових праць VI Міжнародная научно-практическая конференція «Science, innovations and education: problems and prospects» , 13-15 января 2022 года. Токио, Япония, С.58-67, сертифікат.
9. Astistova T. I. Development of electronic settlement system/Т.І. Астістова, D.D Liakhovska, // Збірник наукових праць II Всеукраїнської конференції «Інноватика в освіт, науц та бзнес: виклики та можливост», 18 листопада 2021р., КНУТД, Том 1, С. 157-162
10. Трофименко О.Г., Козін О.Б. Веб-дизайн та HTML-програмування: навч.- метод. посібник. Одеса: Фенікс, 2017. 194 с.
11. The World Bank. World Development Report 2016: Digital Dividends. 2016. p. 330.
12. Efimov Valerii S., Lapteva Alla V. The future of universities: is digitalization the priority? (expert view). Журнал СФУ. 2018. №12.
13. H. Jaakkola, H. Aramo-Immonen, J. Henno and J. Mäkelä, «The digitalization push in universities», 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija. 2016. pp. 1025-1031.
14. P. Linna, T. Mäkinen and H. Keto, «Utilizing MOOCs in the development of education and training programs», 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija. 2016. pp. 861-864.

15. Залізко В. Д., Старинець О. Г., Інноваційні засоби діджиталізації послуг в об'єднаних територіальних громадах. Бізнес Інформ. 2019. №5 (496). С.62-66.
16. Національна стратегія розвитку освіти в Україні на 2015-2021 роки [Електронний ресурс] Режим доступу:  
<http://guonkh.gov.ua/content/documents/16/1517/Attaches/4455.pdf>
17. Державна цільова програма впровадження у навчально-виховний процес загальноосвітніх навчальних закладів інформаційно-комунікаційних технологій "Сто відсотків" на період до 2015 року [Електронний ресурс] Режим доступу:  
<http://guonkh.gov.ua/content/documents/16/1517/Attaches/4455.pdf>
18. Трофименко О.Г., Козін О.Б. Веб-дизайн та HTML-програмування: навч.-метод. посібник. Одеса: Фенікс, 2017. 194 с.
19. 10 безкоштовних інструментів для веб-аналітики. URL: <https://test.ua/2017/01/18/10-free-web-analytics-tool> (дата звернення 14.10.2018).
20. 10 безкоштовних шаблонів на HTML5 і CSS3. URL:  
<http://postovoy.net/10-besplatnyh-shablonov-na-html5-i-css3.html> (дата звернення 14.10.2018).
21. 10 найкращих безкоштовних HTML-редакторів. URL:  
<https://techrocks.ua/2017/09/28/10-best-free-html-editors/> (дата звернення 12.07.2018).
22. 12 huge web design trends for 2018. URL:  
<https://www.creativebloq.com/features/web-design-trends> (accessed 01.12.2018).
23. 15 Web Design Trends to Watch in 2018. URL:  
<https://blog.hubspot.com/marketing/web-design-trends-2017> (accessed 14.09.2018).
24. 19 web design trends for 2018. URL: <https://webflow.com/blog/19-web-design-trends-for-2018> (accessed 18.09.2018).
25. CSS-анімацій з використанням SVG + сайти використовуючи ефект. URL: <http://seo-design.net/css-markup/css-animations-with-svg-websites-uses-it> (дата звернення 11.09.2018).



- 26.9 cutting-edge web design trends for 2018. URL:  
<https://99designs.com/blog/trends/web-design-trends-2018/#mobile-first>  
(accessed 17.09.2018).
27. A brief history of web design for designers. URL:  
<http://blog.froont.com/brief-history-of-web-design-for-designers>  
(accessed 17.09.2018).
28. Биков В., Доменно-фреймова модель педагогічної системи  
[Електронний ресурс]. Режим доступу: <http://www.ime.edu-ua.net/cont/Bykov8.doc>. –Заголовок з екрана.
29. A Complete Guide to Flexbox. URL: <https://css-tricks.com/snippets/css/a-guide-to-flexbox> (accessed 26.07.2018).
30. A vocabulary and associated APIs for HTML and XHTML. W3C. URL:  
<https://www.w3.org/TR/html5> (accessed 14.10.2018).
31. Adaptive vs. Responsive Design. URL: <https://www.interaction-design.org/literature/article/adaptive-vs-responsive-design> (accessed 27.07.2018).
32. Adaptive web design. URL: <https://www.jdvorak.eu/adaptive-web-design/> (accessed 27.07.2018).
33. Best Practices for Speeding Up Your Web Site. URL:  
<https://developer.yahoo.com/performance/rules.html?guccounter=1>  
(accessed 29.09.2019).
34. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014–2019 White Paper. January 30, 2015. URL:  
<https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html> (accessed 26.07.2018).
35. CodePen is a playground for the front end web. URL: <http://codepen.io/>  
(accessed 14.10.2018).
36. CSS Cascading and Inheritance Level 4. URL:  
<https://drafts.csswg.org/css-cascade> (accessed 18.07.2018).
37. CSS Flexible Box Layout Module. URL:  
<https://caniuse.com/#feat=flexbox> (accessed 26.07.2018).

38. CSS Snapshot 2017. URL: <https://www.w3.org/TR/CSS/> (accessed 18.07.2018).
39. CSS Tutorial. URL: [www.w3schools.com/css](http://www.w3schools.com/css) (accessed 14.10.2018).
40. CSS: Вікіпідручник. URL: <http://uk.wikipedia.org/wiki/CSS> (дата звернення 14.10.2018).
41. Cybernetic analytic system. URL: <http://cys.ua> (accessed 28.06.2018).
42. Designing for 10000 Screens: 4 Layout Tips for Responsive Web Design. URL: <http://blog.venturepact.com/designing-for-10000-screens-4-layout-tips-for-responsive-web-design> (accessed 27.07.2018).
43. Flexbugs. URL: <https://github.com/philipwalton/flexbugs#flexbug-9> (accessed 26.07.2018).
44. Gfycat відкриває нові обрії якості GIF завдяки AI. URL: <https://designtalk.club/gfycat-vidkryvaye-novi-obiriyi-yakosti-gif-zavdyaku-ai/> (дата звернення 12.10.2018).
45. Google навчив AI розпізнавати якість фото та їхню «естетичну привабливість». URL: <https://designtalk.club/google-navchuv-ai-rozpoznavaty-yakist-foto-i-navit-yihnyu-estetychnu-pryvablyvist/> (дата звернення 12.10.2018).
46. HTML Colors. URL: [http://www.w3schools.com/html/html\\_colors.asp](http://www.w3schools.com/html/html_colors.asp) (accessed 14.10.2018).
47. HTML Event Attributes. URL: [http://www.w3schools.com/tags/ref\\_eventattributes.asp](http://www.w3schools.com/tags/ref_eventattributes.asp) (accessed 12.07.2018).
48. HTML: Вікіпідручник. URL: <http://uk.wikipedia.org/wiki/HTML> (дата звернення 14.10.2018).
49. Marcotte E. Responsive Web design. URL: <http://www.alistapart.com/articles/responsive-web-design> (accessed 26.07.2018).
50. Official Google Webmaster Central Blog: Rolling out the mobile-friendly update. URL: <https://webmasters.googleblog.com/2015/04/rolling-out-mobile-friendly-update.html> (accessed 26.07.2018).
51. SEO просування, розкрутка та оптимізація сайту в Україні. URL: <http://seoweb.in.ua/services> (дата звернення 29.09.2018).

52. The table element. *HTML5: Edition for Web Authors*. URL: <https://www.w3.org/TR/2011/WD-html5-author-20110809/the-table-element.html> (accessed 25.09.2018).
53. Top 10 Current Web Design Trends for 2018. URL: <https://blog.brightscout.com/top-10-current-web-design-trends-2018> (accessed 14.09.2018).
54. Using Flexbox: Mixing Old and New for the Best Browser Support. URL: <https://css-tricks.com/using-flexbox> (accessed 26.07.2018).
55. WhatFont. URL: <https://chrome.google.com/webstore/detail/whatfont/jaborobgcrjmedljpbcaablplmfcogm> (accessed 14.09.2018).
56. Why Flexboxes Aren't Good for Page Layout. URL: <https://www.xanthir.com/blog/b4580> (accessed 26.07.2018).
57. Адміністрування сайту. URL: <http://webstudio2u.net/ua/studio-web/673-administrirovanie-saita.html> (дата звернення 28.08.2018).
58. Аналіз роботи сайту. URL: <http://webstudio2u.net/ua/design-web/713-analiz-raboty-saita.html> (дата звернення 29.09.2018).
59. Блог. URL: <https://uk.wikipedia.org/wiki/Блог> (дата звернення 28.06.2018).
60. Веб-дизайн сайту і користувацькі помилки. URL: <http://webstudio2u.net/ua/design-web/852-veb-dizain-saita-i-polzovatelskie-oshibki.html> (дата звернення 28.06.2018).
61. Використання CSS flexible-боксів. URL: [https://developer.mozilla.org/uk/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Using\\_CSS\\_flexible\\_boxes](https://developer.mozilla.org/uk/docs/Web/CSS/CSS_Flexible_Box_Layout/Using_CSS_flexible_boxes) (дата звернення 26.07.2018).
62. Використання Flexbox в CSS3 для адаптивного дизайну. URL: <https://sebweo.com/vikoristannya-flexbox-v-css3-dlya-adaptivnogo-dizajnu> (дата звернення 26.07.2018).
63. Електронні видання: довідник. Уклад. Т. Ю. Киричок. К.: НТУУ «КПІ», 2011. 400 с.
64. Етапи розробки сайту. URL: <https://websait.uz.ua/website-development/etapy-rozrobky-sajtu/> (дата звернення 28.08.2018).

65. Етапи створення веб-сайтів. URL: [http://edufuture.biz/index.php?title=Етапи\\_створення\\_веб-сайтів](http://edufuture.biz/index.php?title=Етапи_створення_веб-сайтів) (дата звернення 28.08.2018).
66. Етапи створення веб-сторінок. URL: <http://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/127/127.html> (дата звернення 25.08.2018).
67. Інформаційне наповнення сайту. URL: [https://uk.wikipedia.org/wiki/Інформаційне\\_наповнення\\_сайту](https://uk.wikipedia.org/wiki/Інформаційне_наповнення_сайту) (дата звернення 27.08.2018).
68. Огірко І., Паславська І., Пілат О. Інформаційна система оцінювання якості електронних видань. Вісник Львівського університету. Серія економічна. 2013. Вип. 49. С. 391–398.
69. Огурцов В.В., Гриньов Д.В., Щербаков О.В. Основи веб та веб-дизайн, програмування на боці клієнта: лабораторний практикум з навчальної дисципліни "Веб-технології та веб-дизайн" для студентів напряму підготовки 6.050101 "Комп'ютерні науки". Харків: ХНЕУ ім. С. Кузнеця, 2015. 208 с.
70. Оптимізація сайтів. URL: <http://webstudio2u.net/ua/optimization/97-seo.html> (дата звернення 29.09.2018).
71. Оптимізація сайтів: як покращити готовий сайт? URL: <http://webstudio2u.net/ua/optimization/392-site-optimization.html> (дата звернення 29.09.2018).
72. Філдінг Р.Т. Архитектурні стилі та проектування мережевих архітектур програмного забезпечення /Р.Т. Філдінг, Р.Н. Тейлор, М.С. Акерман, Д.С. Розенблум. - К.: УНІВЕРСИТЕТ КАЛІФОРНІЇ, 2010. – 180 с.
73. Пасічник О.Г., Пасічник О.В., Стеценко І.В. Основи веб-дизайну: навч. посібник. К.: Вид. група ВНУ, 2009. 336 с.
74. Перекладаємо слово респонсивний. URL: <https://slovotvir.org.ua/words/responsyvnyi> (дата звернення 26.07.2018).
75. Пушкар О.І., Климнюк В.Є., Браткевич В.В. Мультимедійні видання: навч. посібник. Харків: Вид. ХНЕУ, 2012. 144 с.

76. Розкрутка сайту. Google Analytics. URL: <http://webstudio2u.net/ua/web-promotion/246-google-analytics.html> (дата звернення 29.09.2018).
77. Розкрутка сайту. Веб-аналітика. URL: <http://webstudio2u.net/ua/web-promotion/245-web-analytics.html> (дата звернення 29.09.2018).
78. Сабін - Вільсон Л. WordPress. Web design. John Wiley & Sons Canada, 2017. 384 с.
79. Тенденції веб-дизайну в 2018 році. URL: <http://webstudio2u.net/ua/design-web/962-tendantsii-veb-dizaina-2018.html> (дата звернення 28.06.2018).
80. Чи може штучний інтелект відібрати роботу дизайнерів? URL: <https://designtalk.club/chy-mozhe-shtuchnyj-intelekt-vidibraty-robotu-u-dyzajneriv/> (дата звернення 12.10.2018).
81. Що таке Flexbox? Опис всіх CSS властивостей, основні принципи, переваги та недоліки. URL: <http://html5.by/blog/flexbox/> (дата звернення 26.07.2018).
82. Що таке SEO аудит і навіщо він потрібен. URL: <http://webstudio2u.net/ua/optimization/689-seo-audit.html> (дата звернення 29.09.2018).
83. Що таке Zero UI, та до чого рухається майбутнє дизайну. URL: <https://designtalk.club/shho-take-zero-ui-do-chogo-ruhayetsya-majbutnye-dyzajnu> (дата звернення 21.09.2018).
84. Що таке юзабіліті сайту і чому це так важливо? URL: <http://webstudio2u.net/ua/faq/50-design/93-useability.html> (дата звернення 20.09.2018).
85. Що таке юзабіліті тестування (юзабіліті аудит). URL: <http://webstudio2u.net/ua/design-web/655-usabiliti-testirovanie.html> (дата звернення 29.09.2018).
86. Як правильно обрати домен. URL: <https://freehost.com.ua/ukr/domain/domainsele> (дата звернення 27.09.2018).
87. Як розпізнати шрифт: підбірка корисних додатків. URL: <https://designtalk.club/yak-rozpiznaty-shryft-pidbirka-korysnyh-dodatkiv/> (дата звернення 12.10.2018).