

**Міністерство освіти і науки України  
Херсонський державний університет  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК, ФІЗИКИ ТА  
МАТЕМАТИКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК  
ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ**

**КОРПОРАТИВНІ ОБЛІКОВІ ЗАПИСИ ЯК ІНСТРУМЕНТ  
ОРГАНІЗАЦІЇ ТА УПРАВЛІННЯ ОСВІТНІМ СЕРЕДОВИЩЕМ  
ЗАКЛАДІВ ВИЩОЇ ОСВІТИ**

**Кваліфікаційна робота (проект)**  
на здобуття ступеня вищої освіти “магістр”

**Виконав:** студент 2-го курсу, 12-241М групи  
Спеціальності: 121 Інженерія програмного  
забезпечення

Освітньо-професійної програми «Інженерія  
програмного забезпечення» другого  
(магістерського) рівня вищої освіти

Кутіщев Володимир Анатолійович

**Керівник** доктор фізико-математичних наук,  
професор Львов Михайло Сергійович

**Рецензент** кандидатка технічних наук,  
доцентка, в.о. завідувача кафедри програмних  
засобів і технологій Херсонського  
національно-технічного університету  
Огнева О.Є.

## ЗМІСТ

<b>ВСТУП</b>	<b>2</b>
<b>РОЗДІЛ 1. КОРПОРАТИВНІ ОБЛІКОВІ ЗАПИСИ В ЗАКЛАДАХ ВИЩОЇ ОСВІТИ</b>	<b>6</b>
1.1. Корпоративні облікові записи в сучасній освіті	6
1.2. Огляд сучасних платформ керування корпоративними обліковими записами в освітніх установах	8
1.3. Сучасний стан використання сервісів корпоративних облікових записів в Херсонському державному університеті.	16
<b>РОЗДІЛ 2. ІНСТРУМЕНТИ РОЗРОБКИ МОДУЛЯ ВІРТУАЛЬНОГО ОСВІТНЬОГО СЕРЕДОВИЩА ЗАКЛАДУ ВИЩОЇ ОСВІТИ</b>	<b>18</b>
2.1. Мова програмування Python та каркас інтернет застосунків Django як основний інструмент із розробки серверної частини модуля.	18
2.2. Мова програмування Javascript та каркас інтернет застосунків React як основний інструмент із розробки клієнтської частини модуля.	28
<b>РОЗДІЛ 3. МОДУЛЬ ОБЛІКУ КОРПОРАТИВНИХ ОБЛІКОВИХ ЗАПИСІВ У ПЛАТФОРМІ KSU24</b>	<b>39</b>
3.1 Вимоги до модуля корпоративних облікових записів у платформі KSU24.	39
3.2 Розробка та впровадження модуля обліку корпоративних облікових записів у платформі KSU24.	40
<b>ВИСНОВКИ</b>	<b>43</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b>	<b>44</b>

## ВСТУП

Сучасна система вищої освіти активно інтегрує інформаційні технології в освітній процес для підвищення ефективності навчання та управління закладами освіти. Одним із ключових аспектів цифровізації освіти є впровадження корпоративних облікових записів, що дозволяють оптимізувати адміністративні та навчальні процеси. Корпоративні акаунти надають можливості для безпечної взаємодії між студентами, викладачами та адміністрацією університету, полегшуючи спільну роботу, управління документами, а також забезпечуючи зручний доступ до навчальних матеріалів. Події останнього десятиріччя, а саме: пандемія COVID-19, повномасштабне військове вторгнення до України з боку Росії, зумовили масштабний перехід до дистанційної форми освіти. За цих умов, для забезпечення безперервності освітнього процесу необхідні сучасні ефективні електронні інструменти. В провідних світових закладах вищої освіти для організації освітнього та адміністративного процесів, використовуються системи обліку корпоративних облікових записи, такі як Microsoft 365, Google Workspace та Zoom. Однак у більшості закладів вищої освіти України процес впровадження подібних рішень стикається із великою кількістю викликів, зокрема через недосконалість технічної інфраструктури та недостатню кваліфікацію персоналу в роботі з новими інформаційними технологіями.

**Актуальність роботи** полягає у необхідності створення системи обліку корпоративних облікових записів у закладі вищої освіти України, для оптимізації процесів адміністративного керування, підвищення ефективності процесів контролю освітнього процесу, та мінімізації негативного впливу від впровадження нових інформаційних технологій у поточні адміністративні процеси закладів.

**Об'єктом** дослідження є корпоративні облікові записи, та комплексні технологічні рішення (засоби для комунікації, управління освітніми матеріалами, захисту даних та ін.), що забезпечують ефективну

взаємодію між здобувачами вищої освіти, науково-педагогічними працівниками та співробітниками відділів адміністрації закладів вищої освіти України.

**Предметом** дослідження є корпоративні облікові записи платформ Microsoft 365, Google Workspace та Zoom, їхні функціональні можливості у якості інструмента організації освітнього процесу у закладах вищої освіти України. Головна увага приділяється впливу цих облікових записів на освітній процес, їхній ролі в підвищенні ефективності навчання, адміністративного управління, а також забезпеченні безпеки та зручності для користувачів.

**Метою** дослідження є створення інформаційної системи керування корпоративними обліковими записами у закладах вищої освіти України для оптимізації процесів організації освітнього середовища та підвищення ефективності адміністративної діяльності.

Згідно до поставленої мети можна виділити наступні **задачі**:

1. Дослідити концепцію корпоративних облікових записів та її використання в освітньому процесі закладів вищої освіти.
2. Проаналізувати переваги та недоліки корпоративних облікових записів на основі платформ Microsoft 365, Google Workspace та Zoom, що використовуються в закладах вищої освіти України.
3. Проаналізувати поточний стан процесів впровадження корпоративних облікових записів у Херсонському державному університеті та оцінити їхній вплив на навчальний процес і адміністративні функції.
4. Розробити вимоги для інформаційної системи моніторингу корпоративних акаунтів.
5. Розробити інтернет застосунок, що дозволяє моніторинг ресурсів.

Для досягнення поставленої мети дослідження були використані наступні наукові методи:

1. **Аналіз і синтез** для дослідження наукових джерел з питань використання корпоративних облікових записів у закладах вищої освіти з метою глибшого розуміння існуючих підходів до організації освітнього процесу за допомогою інформаційних технологій.
2. **Порівняння** для оцінки можливостей і переваг різних систем корпоративних облікових записів, таких як Microsoft 365, Google Workspace та Zoom. Та проведення оцінок використання даних рішень в інших закладах вищої освіти. Цей метод спрямований на виявлення відмінностей між платформами та визначення найефективніших з них для закладів вищої освіти.
3. **Спостереження** за станом процесів впровадження корпоративних облікових записів в Херсонському державному університеті для проведення аналізу реальних даних та практичного досвіду використання цих облікових записів в освітньому процесі та адміністративній діяльності університету.
4. **Вимірювання** рівня задоволеності науково-педагогічних працівників та здобувачів вищої освіти шляхом проведення анкетування відносно використання корпоративних акаунтів у навчанні та комунікації.
5. **Експеримент** відносно впливу від впровадження та використання корпоративних облікових записів на всі аспекти діяльності закладу вищої освіти, зокрема на освітній процес, процеси адміністративного керування та захисту інформації.

**Наукова новизна одержаних результатів** полягає в тому що в теперішній час не розроблені повноцінні методології впровадження

корпоративних облікових записів у закладах вищої освіти України, зокрема у Херсонському державному університеті, зокрема Microsoft 365, Google Workspace та Zoom, та надання оцінки їхнього впливу на організацію освітнього середовища.

**Практичне значення одержаних результатів** полягає в тому, що отримані теоретичні результати та розроблені інструменти керування корпоративними обліковими записами дозволяють оптимізувати процеси адміністрування доступу до корпоративних та освітніх платформ, які залучені до освітнього процесу закладу вищої освіти.

**Апробація результатів дисертації:**

1. Програмний модуль обліку корпоративних облікових записів вже інтегрований та використовується у віртуальному освітньому середовищі KSU24.

2. Доповідь на конференції International Electronic Scientific and Practical Journal “WayScience” (ISSN 2664-4819 (Online))

**Структура роботи.**

Дипломний проект складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

В першому розділі розглядаються теоретичні основи та стан впровадження корпоративних облікових записів в освітній процес закладів вищої освіти України та світу.

В другому розділі описуються основні технології які використовувались під час розробки відповідного програмного модуля для середовища KSU24.

В третьому розділі наведені вимоги користувача даного модуля, інтерфейс користувача модуля, та описано особливості використання даного модуля адміністратором середовища KSU24.

## РОЗДІЛ 1. КОРПОРАТИВНІ ОБЛІКОВІ ЗАПИСИ В ЗАКЛАДАХ ВИЩОЇ ОСВІТИ

### 1.1. Корпоративні облікові записи в сучасній освіті

Хмарні обчислення в даний час, є одним з головних трендів у наданні інтернет-послуг [3]. З точки зору кінцевого користувача, хмарні обчислення використовують різні ІТ-сервіси, програмне забезпечення, дані та апаратні бази в середовищі широкосмугового доступу до мережі незалежно від місця розташування, часу та використовуваного пристрою. Ключовими елементами систем хмарних обчислень є моделі розгортання та послуг. Моделі розгортання визначають зв'язок між апаратною базою хмарних систем і доступним та готовим до використання програмним забезпеченням.

*Моделі розгортання систем хмарних обчислень:*

- модель приватної хмари - послуги надаються лише авторизованим користувачам відповідних організації, що зумовлено необхідністю забезпечення високого рівня безпеки;
- модель Community Cloud – послуги надаються багатьом організаціям, які мають спільні інтереси та які створюють певну спільноту (наприклад: банки, заклади вищої освіти);
- модель загальнодоступної хмари - послуги публічно надаються всім користувачам безкоштовно, або встановлюється фіксована ціна за використання/за користувача через мережу;
- модель гібридної хмари - це поєднання приватної та загальнодоступної хмари, де частина послуг чи застосунків (особливо для обробки конфіденційних бізнес-даних) надається в приватній хмарі.

*Моделі надання послуг визначають, який ІТ-компонент надається як послуга для кінцевого користувача:*

- програмне забезпечення як послуга (SaaS) - користувачеві надається програмне забезпечення, доступне за допомогою веб-браузера або програмного інтерфейсу;
- платформа як послуга (PaaS) - користувачеві надається доступ до різних апаратних частин інформаційно-комунікаційних платформи (операційна система, засоби розробки додатків, засоби баз даних);
- інфраструктура як послуга (IaaS) - користувачеві надається апаратна інфраструктура, мережеві ресурси, сховище та обчислювальна продуктивність.
- зв'язок як послуга (CaaS) - користувачеві надаються такі послуги, як веб-конференція, відеоконференція, інструмент соціальних мереж, обмін миттєвими повідомленнями.

Використання хмарних технологій в освіті зумовлює глобальні зміни в процесах викладання та навчання. Освітні онлайн-курси створюють глобальний віртуальний простір знань та забезпечують підтримку персоналізованого навчання для здобувачів. Електронне навчання загалом стає загальноприйнятим способом отримання освіти що поєднує бездротові мобільні технології, та використовує найсучасніші системи моніторингу для виявлення ситуацій, в яких знаходяться здобувачі, і забезпечення бездоганної адаптивної підтримки поза формальним процесом навчання.

Хмарні обчислення поступово стають частиною віртуальних освітніх середовищ закладів вищої освіти. На думку багатьох дослідників [4], хмарні обчислення – це інфраструктура, яка може привнести нову цінність до систем електронної освіти, оскільки освітні послуги можуть бути доставляється надійним і ефективним способом.

Оскільки учасники освітнього процесу є головним ресурсом будь-якого закладу вищої освіти та ключовим компонентом сучасної системи підтримки освітнього процесу стають програмні рішення для керування людським капіталом. Відповідно зазнають змін і ролі учасників



- освітнього процесу:
- науково-педагогічний працівник перестає бути простим джерелом інформації, та приймає на себе обов'язки радника та екскурсовода в освітньому процесі;
  - здобувач перестає бути пасивним реципієнтом, та переходить до ролі активного учня;

## **1.2.Огляд сучасних платформ керування корпоративними обліковими записами в освітніх установах**

Серед викликів що стоять перед сучасними закладами вищої освіти, одним із головних є задача задоволення зростаючого попиту на цифрову освіту із віддаленим доступом. Технологічні засоби спрямовані на вирішення потреб здобувачів вищої освіти, науково-педагогічних працівників та адміністраторів стають дедалі важливішими, і в теперешній час найбільші ІТ-корпорації світу надають свої відповідні корпоративні програмні рішення.

Google Workspace for Education [1-2], наприклад, надає набір хмарних інструментів продуктивності, які сприяють співпраці, обміну вмістом і спілкуванню в режимі реального часу, а також забезпечують безпеку даних, адаптовану для освітніх середовищ. Подібним чином Microsoft Office 365 розширює традиційне офісне програмне забезпечення хмарними функціями, покращуючи доступність і сприяючи інтерактивному навчанню за допомогою таких інструментів, як Teams. Тим часом роль Zoom у віртуальному навчальному середовищі зробила революцію в онлайн-лекціях і гібридних моделях навчання, пропонуючи функції, які підтримують інтерактивні та масштабовані освітні умови. Крім того, Active Directory (AD) служить основою для безпечного керування обліковими записами в університетських мережах, дозволяючи

адміністраторам ефективно керувати дозволами користувачів, ресурсами та цілісністю даних.

Запущений у 2006 році *Google Apps for Education* – це набір хмарних застосунків, із доступом онлайн через мережу Інтернет, який дозволяє користувачам зберігати, обмінюватися та керувати більшістю своїх даних в Інтернеті, у будь-якому місці, використовуючи практично будь-який пристрій. У той час були хмарні системи були загалом малодоступні, і їхні концепції були відносно в початковому стані, особливо освітні програми [1]. Корпорація Google намагалась зайняти домінуюче місце в освітньому секторі, розробляючи та розгортаючи ці програми в освітніх установах по всьому світі. Після зміни назви в 2019 році на *Google Workspace for Education*, академічний пакет використовується в освітніх закладах з понад 70 мільйонами учнів і вчителів, які його використовують [2].

До складу сервісу “*Google Workspace for Education*” спочатку входило чотири застосунки “*Google Apps*”, що включало сервіс *Google Talk* (на поточний момент не працює), *Календар Google*, “*Сторінка Google*” (на поточний момент сервіс перейменовано в “*Google Sites*”) і система електронної пошти “*Google G-mail*”. Початкова стратегія корпорації *Google* полягала у підключенні закладів шкільної освіти та закладів вищої освіти до цих програм. Також у 2019 році *Google* представив навчальні програми для ведення документів “*Google Docs*”, електронних таблиць “*Google Sheets*”, та застосунки для форм та слайдів. Ці сервіси увійшли до складу серії корпоративних програмних продуктів “*Google for X Edition*”, розділені за окремими категоріями: бізнес, освіта та урядові.

У зв’язку із швидким розвитком Інтернету, компанія *Google* представила сервіс *Google Drive*, який дозволив користувачам зберігати дані і файли в хмарному сховищі. Це дало змогу застосунками конкурувати з кількома популярними хмарними системами системи

зберігання, включаючи Dropbox, OneDrive, Apple iCloud і Box.com.

Ще один великий освітній додаток Google, розроблений після Google Drive - власне програмне забезпечення “LMS Google” та “Google Classroom”. “Google Classroom” є повністю інтегрованим в стандартний пакет Google, що дозволяє установам призначати та збирати матеріали з будь-якої програми Google.

Додаткові переваги сервісу, що в поточний час має назву Google Workspace for Education «рис.1.1.» полягають у тому, що він є практично безкоштовним для будь-якого закладу освіти, та є сумісним і зручним у використанні в усіх застосунках екосистеми Google.

## Google Workspace for Education



*Рис.1.1. Структура Google Workspace for Education у 2022 році.*

Одна із найважливіших переваг системи “Google Classroom LMS” полягає у наданні науково-педагогічним працівникам можливостей для створення онлайн курсів, де вони можуть керувати та інтегрувати практично будь-якими файли. Науково-педагогічні працівники можуть поширювати матеріали з предмету, що викладається, розміщувати додаткові ресурси, призначати завдання та доручення здобувачам, а також виставляти оцінки в режимі реального часу, щоб здобувачі могли відстежувати власний прогрес.

Сервіс “Google Classroom” включає інструменти інтеграції з іншими застосунками “Google Apps” «рис 1.2.». Фактично в системі немає обмежень щодо кількості аудиторій, які може використовувати

користувач [5], також процес створення класних кімнат є відносно простим.

Після створення віртуальної аудиторії, користувачу надається можливість редагування. Після того як здобувачі приєднуються до класу через запрошення, або через код класу, вони отримують можливість взаємодіяти з науково-педагогічними працівниками у “Google Classroom” через безліч відповідних платформ, до складу яких входять загальні оголошення, електронна пошта, приватний чат або коментування дописів чи відповідей.



Рис. 1.2. Загальний вигляд “Google Classroom”

Разом із сервісами для науково-педагогічних працівників система “Google Classroom” надає інструменти для захисту даних та управління користувачами через адміністративну панель “Google Admin Console”. Для отримання доступу студентам необхідно лише отримати відповідний обліковий запис.

*Active Directory* — це служба організації каталогів, розроблена корпорацією Майкрософт для віконної доменної мережі. *Active directory* використовується для зберігання інформації про мережевий ресурс у певному домені «рис.1.3.». Архітектурно “*Active Directory*” є ієрархічною структурою об’єктів. Об’єкти поділяються на три великі категорії: ресурс

(наприклад, принтер), служба (наприклад, електронна пошта), користувач (облікові записи). Сам сервіс “*Active Directory*” надає інформацію про об’єкти, організовує ці об’єкти, контролює доступ і встановлює рівень безпеки. Всередині “*Active Directory*” використовує полегшений протокол доступу до каталогу (LDAP) версії 2 і 3, версію протоколу Microsoft Kerberos і службу доменних імен DNS. “*Active Directory*” дозволяє адміністраторам мережі створювати та керувати доменами, користувачами та об’єктами в мережі. Наприклад адміністратор може створити групу користувачів і надати їм певні права доступу до певних каталогів на сервері. Сервіс “*Active Directory*” був представлений у Windows 2000 як спосіб надання служб каталогів для великих і складних середовищ.

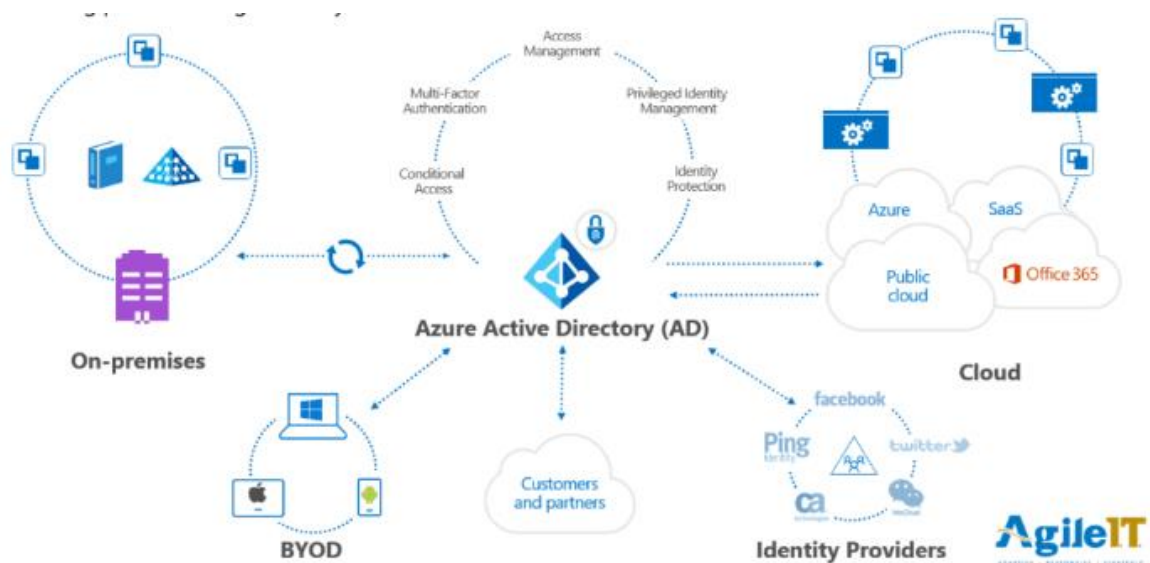


Рис 1.3. Загальна архітектура сервісу “*Active Directory*”

Хмарне рішення Microsoft “*Office 365*” для освіти працює на основі принципу “програмного забезпечення як послуга” (SaaS), і це широка платформа інструментів для спілкування, співпраці, обробки та спільного використання електронних документів і робочих просторів користувачів [6]. Головною перевагою “*Office 365*” є автентифікований доступ користувача з будь-якого місця будь-коли. До стандартного пакету для користувачів, які отримують обліковий запис в “*Office 365*”, входять

онлайн-версії програм Office, OneDrive 1 ТБ пам'яті і 50 ГБ пам'яті електронної пошти, 5 особистих безкоштовних ліцензій для встановлення *Microsoft "Office 365"* «рис 1.4.». З точки зору організації навчального процесу переваги ще більш помітні. Широкий функціонал доступних сервісів разом з уніфікованим доступом для науково-педагогічних працівників і здобувачів вищої освіти створює спільний простір для навчання та співпраці.

Групи для дистанційного навчання можуть бути створені за допомогою сервісу "Outlook", який входить до складу "Microsoft Office 365". Ці групи дозволяють проведення усіх видів дистанційного навчання, починаючи від спільної діяльності, спілкування за допомогою чату, аудіо та відео, онлайн-відвідування, онлайн-іспитів. Спілкування у вигляді відеоконференції може бути організовано через застосунок "Microsoft Teams" [7]. Для онлайн-іспитів використовується застосунок "форми" з пакету Microsoft Office 365.



*Рис.1.4. головна сторінка сервісу "Microsoft office 365 for education"*

Онлайн-іспити проводяться використовуючи різні механізми, зокрема розклади онлайн-іспитів, постановку питань і результати іспитів, а також інші механізми, які полегшують студентам і викладачам адміністрування іспитів. У результатах онлайн-іспитів результати онлайн-тестів видно безпосередньо через комп'ютери кожного студента та викладача. За допомогою сервісу "Microsoft Teams" онлайн-навчання може бути проведення шляхом об'єднання розмов, вмісту і програм в одному місці. Дозволяє науково-педагогічним працівникам організовувати більш креативне навчання, шляхом створення класів для

спільної роботи, об'єднуючись у професійні навчальні спільноти та спілкуючись із колегами-викладачами – все на одній сторінці.

*Платформа “Zoom”* - потужний інструмент для цифрового відео та аудіо зв'язку «рис.1.5.», який використовується переважно для проведення спільних конференцій схожий за функціональністю на застосунок для онлайн спілкування “Skype”. Проте на відміну від “Skype” платформа “Zoom” має певні переваги, які дозволяють класифікувати її як платформу саме для онлайн-освіти. Серед цих переваг можна виділити:

- *живе навчання.* Здобувачі проявляють більше зацікавленості, у випадках коли вони можуть бачити свого вчителя в режимі реального часу. Це також працює в інший спосіб – викладачі відчують характер і стиль навчання своїх студентів, коли вони спілкуються в режимі реального часу.

- *планування.* Платформа надає науково-педагогічним працівникам інструменти для планування занять та освітніх заходів на дні, тижні та будь-які інші часові терміни, передбачені освітньою програмою.

- *відстеження прогресу.* Відстеження академічного прогресу здобувачів є важливою частиною освітнього процесу з точки зору науково-педагогічних працівників. Інструменти, які надають можливості для збору та аналізу досягнень, та своєчасного виявлення проблем, мають суттєве значення для індивідуальної освітньої траєкторії здобувачів. Система відстеження прогресу у “Zoom” працює у фоновому режимі. Увійшовши в адміністративну панель системи відстеження, науково-педагогічні працівники та адміністрація закладу вищої освіти мають можливість помітити, виділити або розглянути шаблони прогресу здобувачів. Це допомагає їм точно визначити сильні та слабкі сторони здобувачів, та надати їм необхідної додаткової допомоги за необхідністю.

- *можливість запису та інтеграції відеоконференцій.* Якісне освітнє програмне забезпечення має бути досить гнучким, для надання науково-педагогічним працівникам інструментів для відстеження прогресу та

формування відповідних звітів, а також дозволяє здобувачам ділитись власними академічними досягненнями.

- *соціальний елемент.* Соціальний елемент освіти дуже важливий, особливо на початковому та середньому рівнях. Хоча немає ідеальної заміни особистому спілкуванню, онлайн-сервіси можуть сприяти таким речам, як обговорення в класі та створення спільноти. В ідеалі соціальні канали слід контролювати, щоб запобігти кібербулінгу.
- *безпека даних.* Якісне освітнє програмне забезпечення та будь-яка онлайн-освітня платформа надає високий пріоритет безпеці даних.



Рис 1.5. Загальний вигляд вікна конференції в “Zoom”

До повсюдного поширення “Zoom” [8], більшість закладів освіти використовували комерційно доступне програмне забезпечення, спрямоване на взаємодію науково-педагогічних працівників та здобувачів вищої освіти, такі як Canvas, Moodle.

Процес переходу від формальної, аудиторної освіти до впровадження платформ “Google Workspace for Education”, “Active Directory”, Microsoft “Office 365”, “Microsoft Teams” та “Zoom” для



деяких освітніх установ стикався із певними труднощами, а саме з необхідністю у отриманні багатьма науково-педагогічними працівниками і здобувачами навичок роботи із даною платформою.

Соціальний аспект навчання був відключений від студентського життя; навчальні програми та планування уроків, оцінювання та інші педагогічні навички та стратегії були приголомшливими для педагогів.

### **1.3. Сучасний стан використання сервісів корпоративних облікових записів в Херсонському державному університеті.**

Хмарні сервіси Google та Microsoft office 365 активно використовуються для організації освітніх процесів у закладах вищої освіти України та надають здобувачам, науково-педагогічним працівникам та співробітникам закладів потужні інструменти для організації процесів обміну інформацією. Головна перевага яку отримує освітній заклад від інтеграції власних освітніх та адміністративних процесів у хмарне середовище через корпоративні облікові записи полягає в об'єднанні функціональних можливостей усіх програмних платформ, які використовуються закладом освіти.

Вступаючи наприклад до Херсонського державного університету здобувач отримує один логін та пароль, який він може використовувати для отримання доступу до будь-яких програмних платформ, які використовує освітній заклад. Центральною платформою в цьому бізнес процесі виступає віртуальне освітнє середовище “KSU24” «рис 1.6».

Підсумовуючи досвід Херсонського державного університету та інших закладів вищої освіти України із впровадження віртуальних освітніх середовищ та корпоративних облікових записів в освітній процес можна виділити наступні аспекти:

- у кризові часи, в умовах дистанційної освіти, теоретичні і практичні

- матеріали не можуть бути повністю засвоєні без керівництва здобувача;
- виходячи з основних труднощів, з якими стикаються здобувачі, науково-педагогічний працівник приймає роль керівника освітнього процесу, приділяючи особливу увагу збалансованості навчального навантаження, враховуючи середній час витрачений на домашнє завдання;
  - негативна сторона освітнього процесу, організованого повністю у дистанційному форматі пов'язана із певними соціальними та психологічними наслідками, що негативно впливають на здобувачів. Головна причина цього – відсутність особистого спілкування та соціалізації, що призводить до демотивації у навчанні та десоціалізації;
  - у першу чергу необхідно сформувати єдине освітнє середовище, зміст якого розробляється в процесі навчання, та яке включає усі аспекти сучасного освітнього процесу у закладах вищої освіти України;
  - за результатами опитування здобувачів та науково-педагогічних працівників, змішана освіта є найефективнішим методом навчання, в сучасних реаліях в Україні та у світі;
  - об'єднання облікових записів на різноманітних платформах, що використовуються в закладі вищої освіти, використовуючи можливості корпоративних облікових записів дозволяє оптимізувати процеси організації дистанційних навчальних занять [8].

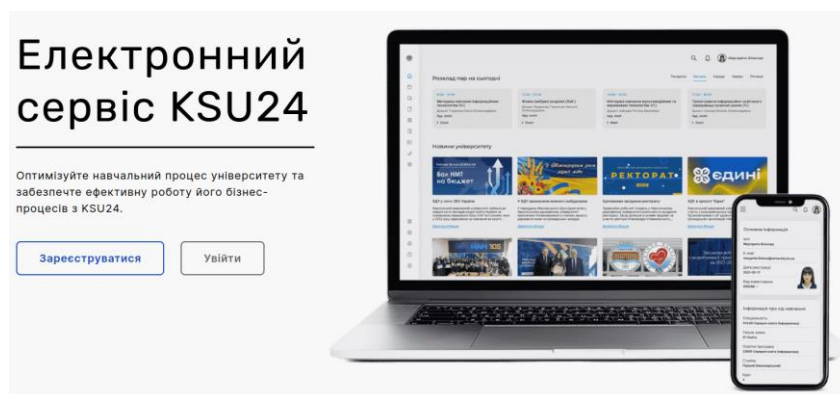


Рис. 1.6. Головне вікно віртуального-освітнього середовища KSU24

## РОЗДІЛ 2. ІНСТРУМЕНТИ РОЗРОБКИ МОДУЛЯ ВІРТУАЛЬНОГО ОСВІТНЬОГО СЕРЕДОВИЩА ЗАКЛАДУ ВИЩОЇ ОСВІТИ

### 2.1. Мова програмування Python та каркас інтернет застосунків Django як основний інструмент із розробки серверної частини модуля.

Python — це мова програмування високого рівня загального призначення. Її філософія спрямована на читабельність програмного коду з використанням значних відступів.[9]

Python є мовою з динамічною типізацією та механізмом збирання сміття, яка підтримує кілька парадигм програмування, включаючи структуроване (зокрема процедурне), об'єктно-орієнтоване та функціональне програмування. Її часто описують як мову «батареї включені» через її повну стандартну бібліотеку.[10][11]

Гвідо ван Россум почав працювати над Python наприкінці 1980-х як наступником мови програмування ABC і вперше випустив її в 1991 році як Python 0.9.0.[12] Python 2.0 був випущений у 2000 році. Python 3.0, випущений у 2008 році, був основною версією, не повністю сумісною з попередніми версіями. Python 2.7.18, випущений у 2020 році, був останнім випуском Python 2.[13]

Python незмінно входить до рейтингу найпопулярніших мов програмування та широко використовується спільнотою машинного навчання.

Python є багато парадигменною мовою програмування яка повністю підтримує парадигми об'єктно-орієнтованого програмування та процедурного програмування, так і багато вбудованих функцій підтримують функціональне та аспектно-орієнтоване програмування (включаючи метапрограмування[13] та мета об'єкти). Python також підтримує багато інших парадигм через можливість встановлення розширень (англ. plugins), наприклад розширення для підтримки

проектування за контрактом[14][15] і логічного програмування [16]. Python відомий як *з'єднувальна мова* [17], яка здатна максимально ефективно працювати із багатьма іншими мовами з легким доступом.

В Python використовується динамічна типізація та комбінація підрахунку посилань разом із прибиральником сміття, що визначає цикл для керування пам'яттю.[18] Він використовує динамічне розпізнавання імен (пізніше зв'язування), яке зв'язує імена методів і змінних на етапі виконання програми.

Архітектура Python пропонує певну підтримку функціонального програмування в стилістиці мови програмування Lisp. Він має функції фільтра, відображення та зменшення; списки розуміння, словники, набори та генераторні вирази.[18] Стандартна бібліотека має два модулі (itertools і functools), які реалізують функціональні інструменти, запозичені з Haskell і Standard ML.[19]

Основна філософія Python коротко викладена в дзен Python [17], до складу якого входять наступні афоризми, такі як:

- *Красиве краще, ніж потворне;*
- *Явне краще, ніж неявне;*
- *Просте краще, ніж складне;*
- *Складне краще, ніж складне;*
- *Читабельність має значення;*

Однак вбудовані функції, які додаються у кожній новій версії Python регулярно порушують ці принципи через що зазнають критики за додавання непотрібного роздуття мови з боку спільноти розробників Python. Відповідь на цю критику полягає в тому, що дзен Python є настановою, а не правилом.[20] Додавання деяких нових функцій було настільки суперечливим, що Гвідо ван Россум пішов у відставку з посади

доброзичливого довічного диктатора після бурхливого ставлення до додавання оператора виразу присвоєння в Python 3.8.

Тим не менш, замість того, щоб вбудувати всю свою функціональність безпосередньо у своє ядро, мову Python було розроблено за концепцією *розширюваності за допомогою модулів*. Ця компактна архітектура зробила мову особливо популярною у якості засобу додавання програмованих інтерфейсів до існуючих програм. Бачення Ван Россума невеликої базової мови з великою стандартною бібліотекою та інтерпретатором, що легко розширюється, впливає з його розчарування в ABC, який підтримував протилежний підхід [18].

За концепцією мови Python стверджується, що головне - це прагнення до простішого, менш захащеного синтаксису та граматики, що надає розробникам можливість вибору методології кодування. На відміну від девізу мови програмування Perl, що визначається як: «є більше, ніж один спосіб зробити це», Python підтримує твердження «повинен бути один — і бажано тільки один — очевидний спосіб зробити це». філософія[17]. На практиці, однак, мова Python пропонує багато способів досягнення того самого завдання. Існує, наприклад, щонайменше три способи форматування рядкового літералу, але немає однозначності щодо того, який з них слід використовувати програмісту.[21] Алекс Мартеллі, співробітник Python Software Foundation і автор книжки про Python, написав: «Описати щось як «розумне» не вважається компліментом у культурі Python».[22]

Розробники Python зазвичай прагнуть уникати передчасної оптимізації та відмовляються від виправлень для некритичних частин еталонної реалізації програмного пакету CPython, які забезпечують незначне збільшення швидкості за рахунок ясності.[23] Швидкість виконання можна покращити, перемістивши критично важливі для швидкості функції до модулів розширення, написаних такими мовами, як C, або використовуючи оперативний компілятор, як-от PyPy. Також

можлива крос-компіляція з іншими мовами, але вона або не забезпечує повного прискорення, яке можна очікувати, оскільки Python є дуже динамічною мовою, або компілюється обмежена підмножина Python, і, можливо, семантика трохи змінено.[22]

Розробники Python прагнуть, щоб ним було цікаво користуватися. Це відображено в його назві — данина поваги британській комедійній групі Monty Python [23] — і в іноді грайливих підходах до посібників і довідкових матеріалів, таких як використання термінів «спам» і «яйця» (посилання на ескіз Monty Python) у прикладах замість часто використовуваних «foo» і «bar»[22][23]. Поширеним неологізмом у спільноті Python є *pythonic*, який має широкий діапазон значень, пов'язаних зі стилем програми. «Pythonic» код може добре використовувати ідіоми Python, бути природним або демонструвати вільне володіння мовою, або відповідати мінімалістичній філософії Python і наголосу на зручності читання. Код, який важко зрозуміти або читається як груба транскрипція з іншої мови програмування, називається не канонічним.[24]

Розробка мовою Python здійснюється в основному за допомогою процесу “Пропозиції вдосконалення Python” (англ. Python Enhancement Proposal, скорочено PEP), основного механізму для пропонування основних нових функцій, збору внесків спільноти щодо проблем і документування проектних рішень Python[25]. Стиль кодування Python розглядається в стандарті PEP8[26]. Виняткові PEP переглядаються та коментуються спільнотою Python та керівною радою[25].

Удосконалення мови відповідає розробці еталонної реалізації технології CPython. Список розсилки *python-dev* є основним форумом для розробки мови. Конкретні питання спочатку обговорювалися в інструменті відстеження помилок Roundup, розміщеному фондом[24]. У 2022 році всі проблеми та обговорення перенесено на GitHub[23]. Спочатку розробка велася на власному сховищі вихідного коду під

керуванням Mercurial, доки Python не перемістився на GitHub у січні 2017 року[24].

Публічні випуски CPython бувають трьох типів, які розрізняються за тим, яка частина номера версії збільшується.

Зворотно несумісні версії, де очікується, що програмний код буде не робочим і його потрібно перенести вручну. Перша частина номера версії збільшується. Такі випуски трапляються рідко — версія 3.0 була випущена через 8 років після 2.0. За словами Гвідо ван Россума, версія 4.0 навряд чи коли-небудь з'явиться[24].

Основні або «функціональні» випуски в основному сумісні з попередньою версією, але містять нові функції. Друга частина номера версії збільшується. Починаючи з Python 3.9, очікується, що ці випуски випускатимуться щорічно[23][24]. Кожна основна версія підтримується виправленнями помилок протягом кількох років після випуску[27].

Випуски виправлень помилок [26], які не вводять нових функцій, відбуваються приблизно кожні 3 місяці та створюються, коли достатню кількість помилок було виправлено на початковому етапі з моменту останнього випуску. У цих випусках також виправляються вразливості системи безпеки. Третя й остання частина номера версії збільшується[26].

Багато альфа-, бета-версій і релізів-кандидатів також випускаються до попереднього перегляду і для тестування перед остаточними випусками. Хоча для кожного випуску існує приблизний графік, вони часто затримуються, якщо код не готовий. Команда розробників Python стежить за станом коду, запускаючи великий набір модульних тестів під час розробки [28].

Головною науковою конференцією з Python є PyCon. Існують також спеціальні наставницькі програми Python, наприклад PyLadies.

У Python 3.12 видалено `wstr`, що означає, що розширення Python [28] потрібно змінити, [29] і 3.10 додано шаблон відповідності до мови [30].

Також у Python 3.12 було вилучено деякі застарілі модулі, і ще більше буде вилучено в майбутньому, застарілим з 3.13; уже застарілий код формату масиву `'u'` видаватиме `DeprecationWarning` з 3.13 і буде видалено в Python 3.16. Натомість слід використовувати код формату `«w»`. Частина `ctypes` також застаріла, і `http.server.CGIHTTPRequestHandler` видасть `DeprecationWarning` і буде видалено в 3.15. Використання цього коду вже має високий потенціал для помилок безпеки та функціональності. Частина модуля введення застаріли, напр. створення класу `typing.NamedTuple` з використанням ключових аргументів для позначення полів і таке (і більше) буде заборонено в Python 3.15.

Синтаксис мови програмування Python — це набір правил, які визначають, як програма Python буде написана та інтерпретована (як системою виконання, так і людьми-читачами) [25]. Мова Python має багато схожості з Perl, C і Java. Однак між мовами є певні відмінності. Python підтримує кілька парадигм програмування, включаючи структуроване, об'єктно-орієнтоване програмування та функціональне програмування, а також може похвалитися системою динамічного типу та автоматичним керуванням пам'яттю.

Синтаксис Python простий і послідовний, дотримуючись принципу: «Повинен бути один — і бажано тільки один — очевидний спосіб зробити це». Мова містить вбудовані типи даних і структури, механізми керування потоком, першокласні функції та модулі для кращого повторного використання та організації коду. Python також використовує англійські ключові слова там, де в інших мовах використовується пунктуація, що сприяє простій візуальній компоновці.



Мова забезпечує надійну обробку помилок за допомогою винятків і включає в себе налагоджувач у стандартній бібліотеці для ефективного вирішення проблем. Синтаксис Python, розроблений для читабельності та простоти використання, що робить його популярним вибором як для початківців, так і для професіоналів [23].

Оскільки Python є мовою з динамічними типами, інформацію про тип несуть значення Python, а не змінні. Усі змінні в Python містять посилання на об'єкти, і ці посилання передаються функціям. Деякі люди (включаючи самого Гвідо ван Россума) назвали цю схему передачі параметрів «викликом за посиланням на об'єкт». Посилання на об'єкт означає ім'я, а передане посилання є «псевдонімом», тобто копією посилання на той самий об'єкт, як у C/C++.

Значення об'єкта можна змінити у викликаній функції за допомогою «псевдоніма», наприклад:

```
>>> alist = ['a', 'b', 'c']

>>> def my_func(al):
...     al.append('x')
...     print(al)
...
>>> my_func(alist)
['a', 'b', 'c', 'x']
>>> alist
['a', 'b', 'c', 'x']
```

Серед мов з динамічною типізацією Python має помірну перевірку типів. Неявне перетворення визначено для числових типів (а також логічних значень), тому можна правильно помножити комплексне число на ціле (наприклад) без явного приведення. Однак немає неявного перетворення між, наприклад, числами та рядками; рядок є недійсним аргументом для математичної функції, яка очікує число.

Python має широкий спектр основних типів даних. Поряд зі звичайними цілими числами та арифметикою з плаваючою комою, він прозоро підтримує арифметику довільної точності, комплексні числа та десяткові числа [27].

Python підтримує широкий спектр операцій з рядками. Рядки в Python є незмінними, тому операція над рядком, наприклад заміна символів, яка в інших мовах програмування може змінити рядок на місці, повертає новий рядок у Python. Проблеми продуктивності іноді вимагають використання спеціальних методів у програмах, які інтенсивно змінюють рядки, наприклад об'єднання масивів символів у рядки лише за потреби.

Одним із дуже корисних аспектів Python є концепція типів колекцій (або контейнерів). Загалом колекція — це об'єкт, який містить інші об'єкти таким чином, що на нього легко посилатися або індексувати. Колекції бувають двох основних форм: послідовності та відображення.

Упорядковані послідовні типи — це списки (динамічні масиви), кортежі та рядки. Усі послідовності позиційно індексуються (від 0 до довжини – 1), і всі, крім рядків, можуть містити будь-який тип об'єкта, включаючи кілька типів в одній послідовності. І рядки, і кортежі незмінні, що робить їх ідеальними кандидатами на ключі словника (див. нижче) [26]. Списки, з іншого боку, є змінними; елементи можна вставляти, видаляти, змінювати, додавати або сортувати на місці.

З іншого боку, відображення — це (часто невпорядковані) типи, реалізовані у формі словників, які «відображають» набір незмінних ключів на відповідні елементи (подібно до математичної функції). Наприклад, можна визначити словник, у якому рядок "toast" відображається на ціле число 42 або навпаки. Ключі в словнику мають бути незмінного типу Python, наприклад, ціле число або рядок, оскільки під капотом вони реалізовані через хеш-функцію. Це значно прискорює час пошуку, але ключі не потрібно змінювати.

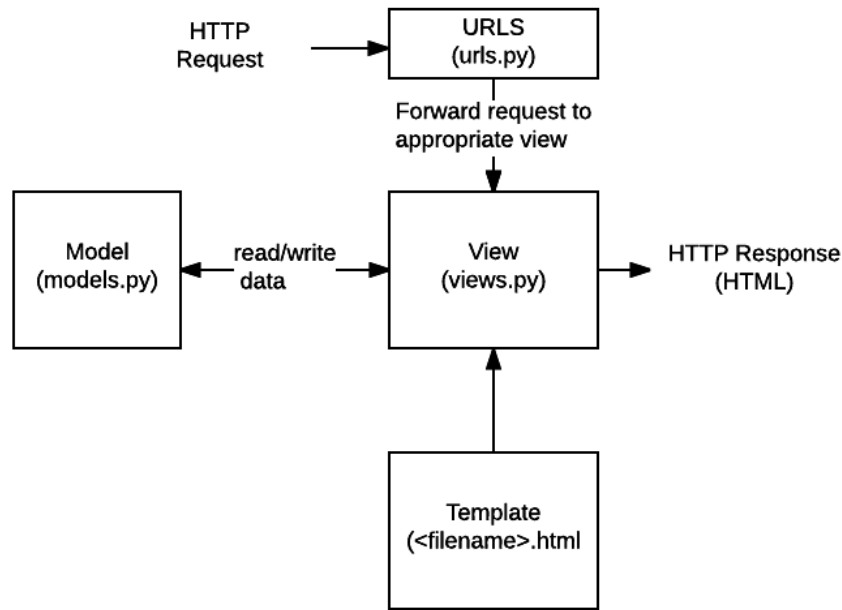
Словники є центральними елементами Python, оскільки вони знаходяться в основі всіх об'єктів і класів: відповідності між іменами змінних (рядками) і значеннями, на які посилаються імена, зберігаються як словники (див. Система об'єктів). Оскільки ці словники доступні безпосередньо (через атрибут `__dict__` об'єкта), метапрограмування є простим і природним процесом у Python.

Тип колекції множин — це не індексована, невпорядкована колекція, яка не містить дублікатів і реалізує такі теоретико-множинні операції, як об'єднання, перетин, різниця, симетрична різниця та перевірка підмножини. Є два типи наборів: `set` і `frozenset`, єдина різниця полягає в тому, що `set` є змінним, а `frozenset` незмінним. Елементи в наборі мають бути хешованими. Таким чином, наприклад, заморожений набір може бути елементом звичайного набору, тоді як навпаки не вірно.

Python також надає широкі можливості маніпулювання колекцією, такі як вбудована перевірка обмеження та загальний протокол ітерації.

У Python все є об'єктом, навіть класи. Класи, як об'єкти, мають клас, який відомий як метаклас. Python також підтримує множинне успадкування та міксини. Мова підтримує широку інтроспекцію типів і класів. Типи можна читати та порівнювати — типи є екземплярами типу. Атрибути об'єкта можна отримати як словник [27]. Оператори можуть бути перевантажені в Python шляхом визначення спеціальних функцій-членів, наприклад, визначення методу з назвою `__add__` для класу дозволяє використовувати оператор `+` для об'єктів цього класу.

Каркас інтернет застосунків Django [31] — це безкоштовна веб-платформа з відкритим кодом на основі Python, яка працює на веб-сервері. Він дотримується архітектурного шаблону модель–шаблон–подання (MTV)[31][32]. Загальний вигляд типового модуля для каркасу інтернет застосунків представлено на «рис 2.1.».



*Рис. 2.1. Загальна структура типового інтернет застосунка, що базується на каркасі Django*

Каркас інтернет застосунків підтримується Django Software Foundation (DSF), незалежною організацією, заснованою в США як некомерційна 501(c)(33).

Головна мета Django — полегшити створення складних веб-сайтів, керованих базами даних. Фреймворк наголошує на повторному використанні та «підключеності» компонентів, меншому коді, низькому зв'язку, швидкому розвитку та принципі «не повторюйся».[32]

Деякі відомі інтернет портали, які побудовані з використанням Django, включають Instagram [35], Mozilla [36], Disqus [37], Bitbucket [38] і Nextdoor [39].

## **2.2. Мова програмування Javascript та каркас інтернет застосунків React як основний інструмент із розробки клієнтської частини модуля.**

Мова програмування JavaScript (скор. JS), є мовою програмування та основною технології Web 2.0 сучасного інтернету, поряд з HTML і CSS.

99% веб-сайтів використовують JavaScript на стороні клієнта для поведінки веб-сторінки [40].

Веб-браузери мають спеціальний вбудований рушій JavaScript, який дозволяє виконання коду клієнта. Ці механізми також використовуються на деяких серверах і різноманітних програмах. Найпопулярнішою системою виконання для небраузерного використання є Node.js.

JavaScript — це мова високого рівня, яка часто скомпільована точно вчасно і відповідає стандарту ECMAScript.[41] Вона має динамічну типізацію, є об'єктно орієнтованою на основі прототипу та містить функції першого порядку. Мова JavaScript є мультипарадигменою, яка підтримує керований подіями, функціональний та імперативний стилі програмування. Мова JavaScript має інтерфейси прикладного програмування (англ. Application Program Interface, скор API ) для роботи з текстом, датами, регулярними виразами, стандартними структурами даних і об'єктною моделлю документа (англ. Document Object Model, скор. DOM).

До стандарту ECMAScript не входять жодні засоби вводу/виводу (I/O), таких як мережеві сховища чи графічні засоби. На практиці веб-браузер або інша система виконання надає JavaScript відповідні інтерфейси прикладного програмування для введення-виведення.

Незважаючи на те, що мови Java і JavaScript подібні за назвою, синтаксисом і відповідними стандартними бібліотеками, ці дві мови відрізняються один від одного і значно відрізняються за архітектурою.

JavaScript займає домінуючу позицію серед мов сценаріїв на стороні клієнта в Інтернеті, і 99% усіх веб-сайтів використовують його для цієї мети.[40] Сценарії вбудовані в документи HTML або включені з них і взаємодіють із об'єктною моделлю документа.

Прикладами поведінки вмісту інтернет сторінок, які містять код мови javascript є:

- завантаження нового вмісту веб-сторінки без перезавантаження сторінки через технології Ajax [41] або WebSocket [42]. Наприклад, користувачі соціальних мереж можуть надсилати й отримувати повідомлення, не залишаючи поточної сторінки;
- анімації веб-сторінок, такі як згасання об'єктів, зміна їх розміру та переміщення;
- гра в браузерні ігри;
- керування відтворенням потокового медіа;
- створення спливаючих оголошень або вікон сповіщень;
- Перевірка вхідних значень веб-форми перед надсиленням даних на сервер;
- реєстрація даних про поведінку користувача та надсилення їх на сервер. Власник веб-сайту може використовувати ці дані для аналітики, відстеження реклами та персоналізації;
- перенаправлення користувача на іншу сторінку;
- зберігання та отримання даних на пристрої користувача за допомогою стандартів зберігання або IndexedDB.

Понад 80% веб-сайтів використовують сторонню бібліотеку JavaScript або веб-фреймворк як частину своїх сценаріїв на стороні клієнта [43].

jQuery, безумовно, є найбільш використовуваним [44]. Інші відомі включають Angular, Bootstrap, Lodash, Modernizr, React, Underscore і Vue[44]. У поєднанні можна використовувати кілька параметрів, наприклад jQuery та Bootstrap [45].

Однак термін «Vanilla JS» було введено для веб-сайтів, які взагалі не використовують бібліотеки чи фреймворки, натомість повністю покладаються на стандартні функції JavaScript [46].

Механізм JavaScript — це програмний компонент, який виконує код JavaScript. Перші движки JavaScript були просто інтерпретаторами, але всі відповідні сучасні движки використовують оперативну компіляцію

для покращення продуктивності [47].

Двигуни JavaScript зазвичай розробляються безпосередньо постачальниками веб-браузерів, і кожен основний браузер має такий. У браузері механізм JavaScript працює разом із механізмом візуалізації через об'єктну модель документа та прив'язки Web IDL[45] Однак використання двигунів JavaScript не обмежується браузерами; наприклад, двигун V8 є основним компонентом системи виконання Node.js [47].

Оскільки специфікація ECMAScript є стандартизованою специфікацією JavaScript, механізм ECMAScript є іншою назвою для цих реалізацій. З появою бібліотеки WebAssembly деякі механізми також можуть виконувати цей код у тій же пісочниці, що й звичайний код JavaScript [45][46].

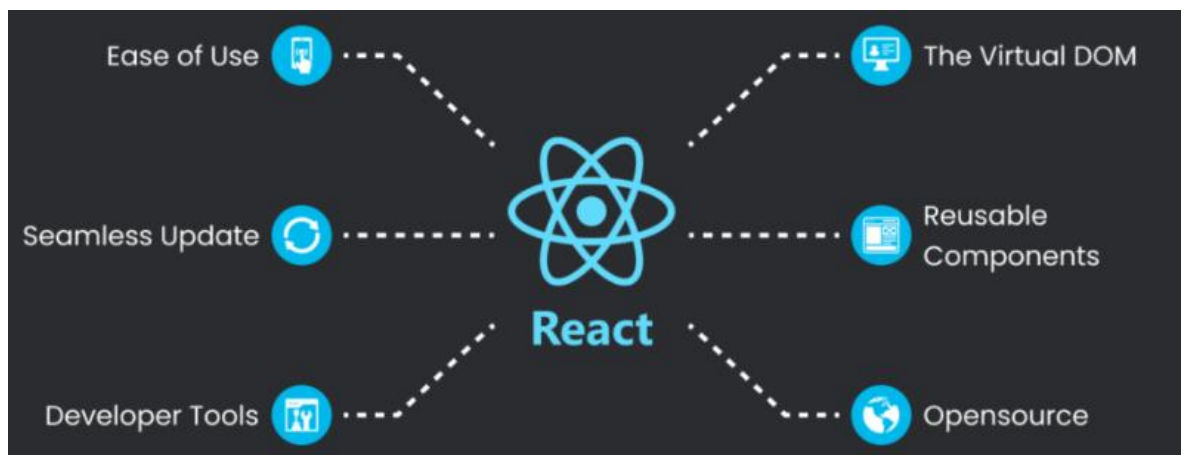
Механізм JavaScript має бути вбудований у систему виконання (наприклад, веб-браузер або окрему систему), для того щоб сценарії могли взаємодіяти із ширшим середовищем. Система виконання включає необхідні API для операцій введення/виведення, таких як мережа, зберігання та графіка, і надає можливість імпортувати сценарії.

JavaScript — це одно потокова мова. Середовище виконання обробляє повідомлення з черги по одному та викликає функцію, пов'язану з кожним новим повідомленням, створюючи кадр стека викликів із аргументами функції та локальними змінними. Стек викликів то зменшується то збільшується відповідно до потреб функції. Коли стек викликів порожній після завершення функції, JavaScript переходить до наступного повідомлення в черзі. Це називається циклом подій, який описується як «запуск до завершення», оскільки кожне повідомлення повністю обробляється перед тим, як розглядається наступне повідомлення. Однак модель паралелізму мови описує цикл подій як неблокуючий: програмний ввід/вивід виконується за допомогою подій і функцій зворотного виклику. Це означає, наприклад, що JavaScript може обробляти клацання миші, очікуючи, поки запит до бази даних поверне

інформацію [45]. Відомими автономними середовищами виконання є Node.js, Deno та Bun.

React (також відомий як React.js або ReactJS) — це безкоштовна інтерфейсна бібліотека JavaScript із відкритим вихідним кодом[48][49] для створення інтерфейсів користувача на основі компонентів від Facebook Inc. Підтримується Meta (раніше Facebook) і спільнота окремих розробників і компаній [50].

React можна використовувати для розробки односторінкових, мобільних або серверних додатків із такими фреймворками, як Next.js. Оскільки React займається лише інтерфейсом користувача та відтворенням компонентів у об'єктну модель документа, програми React часто покладаються на бібліотеки для маршрутизації та інших функцій на стороні клієнта.[9][10] Ключовою перевагою React є те, що він повторно рендерить лише ті частини сторінки, які змінилися, уникаючи непотрібного повторного рендерингу незмінених елементів об'єктної моделі документа. Вперше він був запущений 29 травня 2013 року [52].



*Рис.2.2. Загальні функціональні можливості каркасу інтернет застосунків React*

Серед основних функціональних можливостей React, які в загальному вигляді представлені на «рис 2.2.» особливо виділяються наступні:

- *декларативність*. React дотримується парадигми декларативного



програмування. Розробники проектують представлення для кожного стану програми, і React оновлює та рендерить компоненти, коли дані змінюються. Це відрізняється від імперативного програмування;

- *компонентність*. Код React складається із сутностей, які називаються компонентами. Ці компоненти є модульними та багаторазово використовуваними. Застосунки React зазвичай складаються з багатьох рівнів компонентів. Компоненти рендеряться в кореневий елемент у об'єктній моделі документа за допомогою бібліотеки ReactDOM. Під час візуалізації компонента значення передаються між компонентами через властивості (скорочення від «властивості»). Внутрішні значення компонента називаються його станом [52]. Функціональні компоненти оголошуються за допомогою функції (використовуючи синтаксис функції JavaScript або виразу функції зі стрілкою), яка приймає один аргумент "props" і повертає відповідь у форматі JSX. Починаючи з React v16.8, функціональні компоненти можуть використовувати стан за допомогою “хука” “useState”.
- *серверні компоненти React* (англ. React Server Components) [53] — це функціональні компоненти, які працюють виключно на сервері. Концепція була вперше представлена в доповіді "Отримання даних за допомогою серверних компонентів"[53]. Хоча концепція подібна до візуалізації на стороні сервера, RSC не надсилають відповідний JavaScript клієнту, оскільки не відбувається гідратації. Наразі серверні компоненти частіше за все використовують з каркасом інтернет застосунків Next.js. Однак вони можуть бути асинхронними функціями, що дозволяє їм безпосередньо виконувати асинхронні операції:

*асинхронна функція MyComponent() {*

*const message = await fetchMessageFromDb();*

*повернення (*

```
<div>Повідомлення: {message}</div>
```

```
);
```

```
}
```

- *маршрутизація*. Сам каркасом інтернет застосунків “React” не має вбудованої підтримки маршрутизації. Каркас інтернет застосунків “React” — це насамперед бібліотека для створення користувальницьких інтерфейсів, і вона не включає повноцінне рішення маршрутизації з коробки. Бібліотеки сторонніх розробників можна використовувати для обробки маршрутизації в програмах каркасу інтернет застосунків React [52]. Це дозволяє розробнику визначати маршрути, керувати навігацією та обробляти зміни URL-адрес у зручний для React спосіб;
- *віртуальна об’єктна модель документа*, яка використовується для реалізації справжньої об’єктної моделі документа. Каркас інтернет застосунків React створює в кеш-пам’яті структури даних, обчислює отримані відмінності, а потім ефективно оновлює реальну віртуальну об’єктну модель документа, що відображається у браузері [53]. Цей процес називається звіркою. Це дозволяє програмісту писати код так, ніби вся сторінка рендериться при кожній зміні, тоді як React рендерить лише компоненти, які фактично змінюються;
- *візуалізація на стороні сервера* (англ. Server Side Rendering) відноситься до процесу візуалізації клієнтської програми JavaScript на сервері, а не в браузері. Це може покращити продуктивність програми, особливо для користувачів із повільним з’єднанням. За допомогою технології візуалізація на стороні сервера початковий HTML-код, який надсилається клієнту, включає повністю відтворений інтерфейс користувача програми. Це дозволяє браузеру клієнта негайно відображати інтерфейс користувача, замість того, щоб чекати, поки JavaScript завантажиться та запуститься перед відтворенням інтерфейсу користувача. Каркас інтернет застосунків React підтримує візуалізацію на стороні сервера, що дозволяє розробникам відтворювати компоненти

каркасу інтернет застосунків React на сервері та надсилати отриманий HTML клієнту. Це може бути корисним для покращення продуктивності програми, а також для оптимізації пошукової системи.

### **1.3. База даних PostgreSQL.**

Postgres — це безкоштовна система керування реляційними базами даних (англ. Relational Database Management Systems) із відкритим вихідним кодом, яка наголошує на принципах розширюваності та сумісності з стандартом SQL. PostgreSQL містить транзакції з властивостями атомарності (англ. atomicity), узгодженості (англ. consistency), ізоляції (англ. isolation), довговічності (англ. durability) (ACID), що дозволяє автоматично оновлювати подання, матеріалізовані подання, тригери, зовнішні ключі та збережені процедури.[54] Він підтримується в усіх основних операційних системах, включаючи Windows, Linux, macOS, FreeBSD і OpenBSD, і обробляє низку робочих навантажень від окремих машин до сховищ даних, озер даних [55] або веб-сервісів із багатьма одночасними користувачами.

Глобальна група розвитку PostgreSQL зосереджена лише на розробці механізму баз даних і тісно пов'язаних компонентів. Технічно це ядро становить сам PostgreSQL, але існує велика спільнота розробників та екосистема, яка надає інші важливі набори функцій, які традиційно можуть надаватися постачальником пропрієтарного програмного забезпечення. До них належать функції механізму бази даних спеціального призначення, такі як ті, які необхідні для підтримки геопросторової [55] або часової [56] бази даних або функції, які емулюють інші продукти баз даних [57]. Крім того, від третіх сторін доступні різноманітні функції інтерфейсу користувача та машини, такі як графічні інтерфейси користувача [58] або набори інструментів для балансування навантаження та високої доступності [59]. Велика стороння мережа підтримки PostgreSQL, що складається з людей, компаній, продуктів і

проектів, хоча й не є частиною The PostgreSQL Development Group, є важливою для впровадження та використання механізму баз даних PostgreSQL і складають велику екосистему PostgreSQL [60].

Було проведено багато неформальних досліджень продуктивності PostgreSQL [59]. Покращення продуктивності, спрямовані на покращення масштабованості, почалися значною мірою з версії 8.1. Прості тести між версією 8.0 і версією 8.4 показали, що остання була більш ніж у десять разів швидшою на робочих навантаженнях лише для читання та принаймні в 7,5 разів швидшою як на робочих навантаженнях читання, так і запису [58].

Перший галузевий стандарт і аналогічний тест був завершений у червні 2007 року з використанням Sun Java System Application Server (власна версія GlassFish) 9.0 Platform Edition, сервера Sun Fire на базі UltraSPARC T1 і PostgreSQL 8.2.[57] Цей результат 778.14 SPECjAppServer2004 JOPS@Standard вигідно відрізняється від 874 JOPS@Standard з Oracle 10 у системі HP-UX на базі Itanium [60]. У серпні 2007 року Sun представила покращену оцінку тесту 813,73 SPECjAppServer2004 JOPS@Standard. Завдяки зниженій ціні тестованої системи співвідношення ціна/продуктивність покращилося з 84,98 дол. США за JOPS до 70,57 дол. США за JOPS [59].

Конфігурація PostgreSQL за замовчуванням використовує лише невелику кількість виділеної пам'яті для критичних для продуктивності цілей, таких як кешування блоків бази даних і сортування. Це обмеження пов'язане насамперед із тим, що старіші операційні системи потребували змін у ядрі, щоб дозволити розподіл великих блоків спільної пам'яті [57]. PostgreSQL.org надає поради щодо базової рекомендованої практики продуктивності у вікі [56].

У квітні 2012 року Роберт Хаас з EnterpriseDB продемонстрував лінійну масштабованість ЦП PostgreSQL 9.2 за допомогою сервера з 64 ядрами [57].

Matloob Khushi виконав порівняльний аналіз між PostgreSQL 9.0 і MySQL 5.6.15 щодо їх здатності обробляти геномні дані. У своєму аналізі продуктивності він виявив, що PostgreSQL витягує геномні області, що перекриваються, у вісім разів швидше, ніж MySQL, використовуючи два набори даних по 80 000 кожен, утворюючи випадкові ділянки людської ДНК. Вставка та завантаження даних у PostgreSQL також були кращими, хоча загальні можливості пошуку в обох базах даних були майже еквівалентними [58].

PostgreSQL спочатку називався POSTGRES, посилаючись на його походження як наступника бази даних Ingres, розробленої в Каліфорнійському університеті в Берклі. У 1996 році проект було перейменовано в PostgreSQL, щоб відобразити підтримку SQL. Після перегляду в 2007 році команда розробників вирішила зберегти назву PostgreSQL і псевдонім Postgres.

PostgreSQL підтримує механізми керування паралелізмом за допомогою багатOVERСІЙНОГО керування паралелізмом, який надає кожній транзакції «моментальний знімок» бази даних, що дозволяє вносити зміни, не впливаючи на інші транзакції. Це значною мірою усуває потребу в блокуванні читання та гарантує, що база даних підтримує принципи ACID. PostgreSQL пропонує чотири рівні ізоляції транзакцій: читання без фіксації, читання з фіксацією, повторюване читання та серіалізація. Оскільки PostgreSQL несприйнятливий до брудного читання, запит на рівень ізоляції транзакції Read Uncommitted надає замість цього здійснене читання. PostgreSQL підтримує повну можливість серіалізації за допомогою методу серіалізованої ізоляції моментальних знімків (SSI) [59] Реалізація PostgreSQL схильна до проблем з продуктивністю, які вимагають налаштування під час великого навантаження на запис, який оновлює існуючі рядки [60].

PostgreSQL має вбудовану підтримку механізмів реплікацій, засновану на асинхронній доставці змін (журналів попереднього запису)

до вузлів реплік із можливістю виконувати запити лише для читання до цих реплікованих вузлів. Це дозволяє ефективно розподіляти трафік читання між кількома вузлами. Попереднє програмне забезпечення реплікації, яке дозволяє подібне масштабування читання, зазвичай покладалося на додавання тригерів реплікації до основного, збільшуючи навантаження.

PostgreSQL включає вбудовану синхронну реплікацію[60], яка гарантує, що для кожної транзакції запису головний пристрій чекає, доки принаймні один вузол-реплікатор не почне процедуру запису даних до свого журналу транзакцій. На відміну від інших систем баз даних, довговічність транзакції (незалежно від того, чи є вона асинхронною чи синхронною) може бути визначена для бази даних, для кожного користувача, для кожного сеансу або навіть для окремої транзакції. Це може бути корисним для робочих навантажень, які не вимагають таких гарантій, і може бути необхідним для всіх даних, оскільки це сповільнює продуктивність через вимогу підтвердження транзакції, що досягає синхронного режиму очікування.

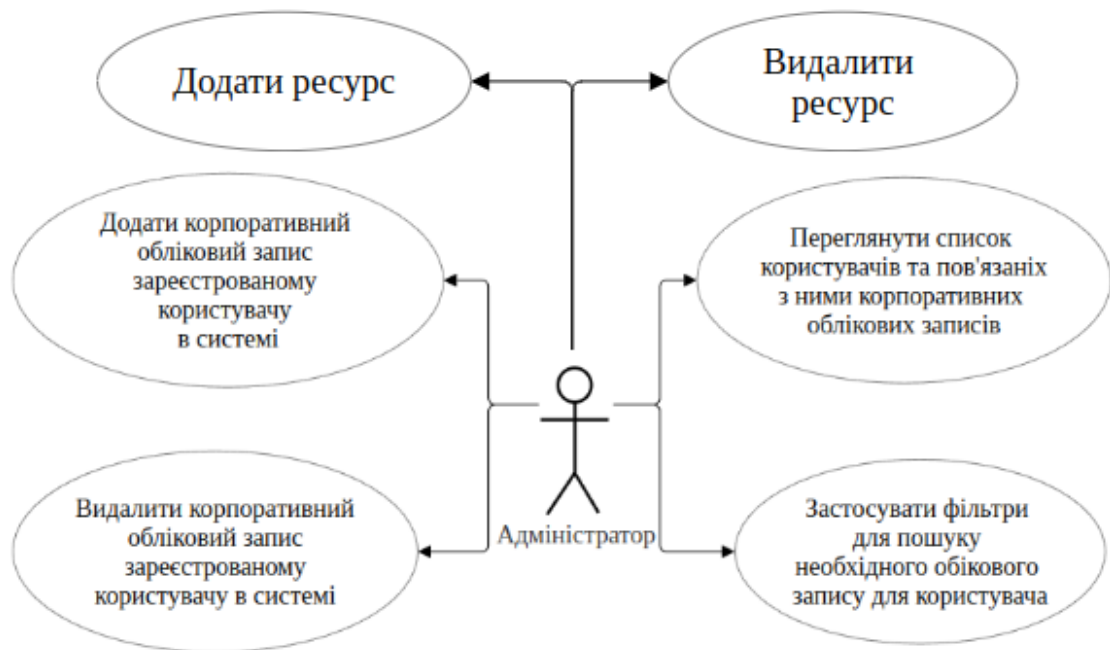
Резервні сервери можуть бути синхронними і асинхронними. Синхронні резервні сервери можна вказати в конфігурації, яка визначає, які сервери є кандидатами для синхронної реплікації.

Механізм синхронної реплікації із кількома головними не включена в стандартне ядро PostgreSQL. Postgres-XC, який базується на PostgreSQL, забезпечує масштабовану синхронну реплікацію з кількома головними [59].

## РОЗДІЛ 3. МОДУЛЬ ОБЛІКУ КОРПОРАТИВНИХ ОБЛІКОВИХ ЗАПИСІВ У ПЛАТФОРМІ KSU24

### 3.1 Вимоги до модуля корпоративних облікових записів у платформі KSU24.

Система обліку корпоративних облікових записів користувачів у віртуальному освітньому середовищі Херсонського державного університета розробляється у вигляді окремого програмного модуля для платформи KSU24, для ролі головного адміністратора системи. Функціональні можливості даного модуля представлені на діаграмі прецедентів «рис 3.1.»



*Рис. 3.1. Діаграми прецедентів для модуля обліку корпоративних облікових записів для ролі адміністратора*

Даний модуль доступний лише в адміністративній панелі системи KSU24, та тільки для ролі користувача "Адміністратор". Як видно із діаграми на «рис 3.1.» модуль призначений для моніторингу поточного стану рівнів доступу користувачів, які мають обліковий запис на платформі KSU24, до корпоративних систем (позначені як "ресурс" на

схемі). Функціональні можливості модуля включають базові операції додавання, перегляду, редагування та видалення (англ. create, read, update, delete скор. CRUD) які адміністратор може виконувати із сутностями корпоративних систем до яких користувачі можуть отримати доступ, і записами про відповідні доступи самих користувачів до конкретної корпоративної системи.

### 3.2 Розробка та впровадження модуля обліку корпоративних облікових записів у платформі KSU24.

З точки зору розробника система обліку корпоративних облікових записів розроблено як типовий модуль, для адміністративної панелі KSU24. Зовнішній вигляд модуля представлено на «рис. 3.2.».

<input type="checkbox"/>	ДОСТУП ОСОБИ ДО РЕСУРСУ	СТВОРЕНО	MODIFIED	DELETED AT	ERRORS LEVEL	PERSON	ACCESS POINT	STATUS	COMMENT
<input type="checkbox"/>	Кутішев Володимир Анатолійович   Udemu	22 жовтня 2024 р. 17:42	22 жовтня 2024 р. 17:42	-	OK	Кутішев Володимир Анатолійович (1981-07-31)   8832210	Udemu   https://ua.udemy.com/	●	Через левачука у-нету
<input type="checkbox"/>	Кутішев Володимир Анатолійович   Microsoft 365	22 жовтня 2024 р. 17:33	22 жовтня 2024 р. 17:33	-	OK	Кутішев Володимир Анатолійович (1981-07-31)   8832210	Microsoft 365   https://www.office.com/	●	Адміністратор
<input type="checkbox"/>	Кутішев Володимир Анатолійович   Zoom	22 жовтня 2024 р. 17:32	22 жовтня 2024 р. 17:32	-	OK	Кутішев Володимир Анатолійович (1981-07-31)   8832210	Zoom   https://zoom.us/	●	Адміністратор
<input type="checkbox"/>	Кутішев Володимир Анатолійович   Google Workspace	22 жовтня 2024 р. 17:31	22 жовтня 2024 р. 17:31	-	OK	Кутішев Володимир Анатолійович (1981-07-31)   8832210	Google Workspace   https://workspace.google.com	●	Адміністратор
<input type="checkbox"/>	Мартос Андрій Сергійович   Головний сайт	22 жовтня 2024 р. 13:19	22 жовтня 2024 р. 13:19	-	OK	Мартос Андрій Сергійович (1987-08-28)   temp11111	Головний сайт   https://www.krsu.edu/	●	Головний адміністратор

Рис.3.2. Зовнішній вигляд модуля обліку корпоративних облікових засобів

Для простоти та ефективності інформація представлена у вигляді таблиці де адміністратор бачить необхідну інформацію про зареєстрованого користувача:

- обліковий запис на відповідній платформі, на «рис.3.2.» представлені наступні платформи: Udemu, Microsoft 365, Zoom, Google Workspace;
- інформація про точний час та дату, коли користувачеві було надано доступ до відповідного ресурсу, та коли рівні доступу було змінено;
- інформація про те чи є даний запис активним у поточний час;



- коментар, який містить додаткову інформацію про конкретний корпоративний обліковий запис, від адміністратора.

Для надання доступу до конкретного ресурса, користувачу зареєстрованому в системі адміністратор натискає на кнопку “додати доступ до ресурсу”, яка розташована в правому верхньому кутку головної сторінки модуля обліку корпоративних облікових засобів, після чого потрапляє на відповідну сторінку додавання «рис. 3.3».

Person:  + add  
Кутіщев Володимир Анатолійович (1981-07-31) | 8832210  
Людина

Access point:    
Ресурс доступу

Status  
Чи доступ є активним

Comment: 

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip

  
Коментар

Id: 4c73baac-dc5d-4cba-a71e-9f9d587ab051

Modified: 30 жовтня 2024 р. 09:45

Створено: 30 жовтня 2024 р. 09:45

Deleted at: -

*Рис. 3.3. Зовнішній вигляд сторінки “Додати доступ до ресурсу”*

На цій сторінці адміністратор вводить всю необхідну інформацію, деякі поля будуть заповнені автоматично, наприклад поле “id”, яке містить ідентифікатор з бази даних системи “KSU24”, та поля що містять інформацію про час коли доступ було додано, або змінено.

Для керування власно ресурсами корпоративних систем, до яких користувачам може бути надано доступ створена окрема сторінка для адміністратора «рис.3.4.»

Домівка > Адміністрування доступу > Ресурси доступу

Виберіть Ресурс доступу щоб змінити

ДОДАТИ РЕСУРС ДОСТУЛУ

ВІДФИЛЬТРУВАТИ

За errors level: Всі

За deleted at: Всі

2024 22 Жовтень

Дія: [.....] Вперед Обрано 0 з 6

РЕСУРС ДОСТУЛУ	СТВОРЕНО	МОДИФІКОВАНО	DELETED AT	ERRORS LEVEL	NAME	URL	DESCRIPTION
<input type="checkbox"/> Синтезія   <a href="https://www.synthesia.io/">https://www.synthesia.io/</a>	22 жовтня 2024 р. 17:41	22 жовтня 2024 р. 17:41	-	OK	Синтезія	<a href="https://www.synthesia.io/">https://www.synthesia.io/</a>	
<input type="checkbox"/> Udemy   <a href="https://ua.udemy.com/">https://ua.udemy.com/</a>	22 жовтня 2024 р. 17:34	22 жовтня 2024 р. 17:34	-	OK	Udemy	<a href="https://ua.udemy.com/">https://ua.udemy.com/</a>	
<input type="checkbox"/> Zoom   <a href="https://zoom.us/">https://zoom.us/</a>	22 жовтня 2024 р. 17:29	22 жовтня 2024 р. 17:29	-	OK	Zoom	<a href="https://zoom.us/">https://zoom.us/</a>	
<input type="checkbox"/> Google Workspace   <a href="https://workspace.google.com">https://workspace.google.com</a>	22 жовтня 2024 р. 17:28	22 жовтня 2024 р. 17:28	-	OK	Google Workspace	<a href="https://workspace.google.com">https://workspace.google.com</a>	
<input type="checkbox"/> Microsoft 365   <a href="https://www.office.com/">https://www.office.com/</a>	22 жовтня 2024 р. 17:27	22 жовтня 2024 р. 17:27	-	OK	Microsoft 365	<a href="https://www.office.com/">https://www.office.com/</a>	
<input type="checkbox"/> Головний сайт   <a href="https://www.kspu.edu/">https://www.kspu.edu/</a>	22 жовтня 2024 р. 13:19	22 жовтня 2024 р. 13:19	-	OK	Головний сайт	<a href="https://www.kspu.edu/">https://www.kspu.edu/</a>	Сайт kspu.edu Контроль доступу редакторів

*Рис. 3.4. Сторінка перегляду корпоративних систем до яких користувач платформи “KSU24” може отримати доступ*

На даній сторінці виводиться перелік корпоративних систем вже зареєстрованих в платформі KSU24, дата і час коли перший користувач отримав доступ до конкретної платформи, та коли відповідний запис було відредаговано, адреса головної сторінки. Так само як і на загальній сторінці модуля кнопка для додавання нової платформи розташована у правому верхньому кутку сторінки.

## ВИСНОВКИ

В результаті виконання даної роботи було розроблено модуль для обліку корпоративних облікових записів у платформі KSU24.

Було проаналізовано переваги та недоліки корпоративних облікових записів у платформах Microsoft 365, Google Workspace та Zoom, що використовуються в закладах вищої освіти України, та зокрема у Херсонському державному університеті.

Був проаналізувати поточний стан процесів впровадження корпоративних облікових записів у Херсонському державному університеті та на основі цього аналізу розроблено вимоги користувача для модуля обліку у платформі KSU24.

Було розроблено модуль обліку корпоративних облікових акаунтів для платформи KSU24, використовуючи мову програмування Python, каркас інтернет застосунків Django, для серверної частини модуля, та мову програмування JavaScript та каркас інтернет застосунків React для клієнтської частини модуля.

Даний модуль вже розміщено на основному сервері системи KSU24, та може бути використаний користувачами платформи з рівнем доступу “Адміністратор”.

Напрямами подальшого дослідження за цією темою є вдосконалення модуля обліку корпоративних облікових акаунтів шляхом додавання функцій для статистичної обробки інформації, що відображається в даному модулі, та функцій для генерації звітної документації.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Vouk, M. A. Cloud computing; Issues, research and implementations. ITI 2008, (pp. 31–40)
2. Fenton, W.. Google Classroom could bridge a gap in online learning. PC Magazine. 2017/
3. Ray, Partha Pratim (2018). "An Introduction to Dew Computing: Definition, Concept and Implications - IEEE Journals & Magazine". *IEEE Access*. **6**: 723–737. doi:10.1109/ACCESS.2017.2775042. ISSN 2169-3536. S2CID 3324933
4. Montazerolghaem, Ahmadreza; Yaghmaee, Mohammad Hossein; Leon-Garcia, Alberto (September 2020). "Green Cloud Multimedia Networking: NFV/SDN Based Energy-Efficient Resource Allocation". *IEEE Transactions on Green Communications and Networking*. **4** (3): 873–889. doi:10.1109/TGCN.2020.2982821. ISSN 2473-2400. S2CID 216188024.
5. Shankland, Stephen (October 1, 2014). "Google to offer schools, students unlimited storage for free". *CNET*. CBS Interactive. Retrieved February 21, 2017.
6. Grindei Laura, Orza Bogdan, Ciupe Aurelia, Meza Serban. University Learning Management System based on Office 365. Faculty of Electrical Engineering. Technical University of Cluj-Napoca, Cluj-Napoca
7. Foley, Mary Jo (November 2, 2016). "Microsoft launches its Slack competitor, Microsoft Teams".
8. Yu. Krylova-Grek, M. Shyshkina, "Blended Learning Method for Improving Students' Media Literacy Level", Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer, Kharkiv, Ukraine, October 06-10, 2020. pp. 1272-1285. (in English)

9. Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises"
10. "About Python". Python Software Foundation. Archived from the original on 20 April 2012. Retrieved 24 April 2012., second section "Fans of Python use the phrase "batteries included" to describe the standard library, which covers everything from asynchronous processing to zip files."
11. "PEP 206 – Python Advanced Library". *Python.org*. Archived from the original on 5 May 2021. Retrieved 11 October 2021.
12. Rossum, Guido Van. "The History of Python: A Brief Timeline of Python". *The History of Python*. 20 January 2009
13. Peterson Benjamin. "Python 2.7.18, the last release of Python 2". *Python Insider*. 20 April 2020
14. "PyDBC: method preconditions, method postconditions and class invariants for Python". Archived from the original on 23 November 2019. Retrieved 24 September 2011.
15. "Contracts for Python". Archived from the original on 15 June 2020. Retrieved 24 September 2011.
16. "PyDatalog". Archived from the original on 13 June 2020. Retrieved 22 July 2012.
17. "Glue It All Together With Python". *Python.org*. Retrieved 30 September 2024.
18. "Extending and Embedding the Python Interpreter: Reference Counts". *Docs.python.org*.
19. Hettinger, Raymond. "PEP 289 – Generator Expressions". *Python Enhancement Proposals*. 30 January 2002. Python Software Foundation.
20. "Confusion regarding a rule in The Zen of Python". *Python Help - Discussions on Python.org*. 3 May 2022.

21. Bader, Dan. "Python String Formatting Best Practices". *Real Python*. Archived from the original on 18 February 2024. Retrieved 25 February 2024.
22. Martelli, Alex; Ravenscroft, Anna; Ascher, David (2005). *Python Cookbook, 2nd Edition*. O'Reilly Media. p. 230. ISBN 978-0-596-00797-3. Archived from the original on 23 February 2020. Retrieved 14 November 2015.
23. Downey, Allen B. (May 2012). *Think Python: How to Think Like a Computer Scientist* (version 1.6.6 ed.). Cambridge University Press. ISBN 978-0-521-72596-5.
24. "Code Style – The Hitchhiker's Guide to Python". *docs.python-guide.org*.
25. Warsaw, Barry; Hylton, Jeremy; Goodger, David (13 June 2000). "PEP 1 – PEP Purpose and Guidelines". *Python Enhancement Proposals*. Python Software Foundation.
26. "PEP 602 – Annual Release Cycle for Python".
27. Norwitz, Neal (8 April 2002). "[Python-Dev] Release Schedules (was Stability & change)"
28. "Python Buildbot". *Python Developer's Guide*. Python Software Foundation.
29. "PEP 623 – Remove “wstr” from Unicode". *Python.org*.
30. "PEP 634 – Structural Pattern Matching: Specification". *Python.org*.
31. "FAQ: General - Django documentation - Django". Retrieved 30 April 2016.
32. "FAQ: General - Django documentation - Django". Retrieved 30 April 2016.
33. Adrian Holovaty, Jacob Kaplan-Moss; et al. *The Django Book*.

34. JavaScript [Электронный ресурс]: статья. – Режим доступа:<http://ru.wikipedia.org/wiki/JavaScript> .
35. Instagram[Электронный ресурс]: статья. – Режим доступа:  
<https://www.instagram.com/>
36. Mozilla [Электронный ресурс]: статья. – Режим доступа:  
<https://www.mozilla.org/uk/>
37. Disqus [Электронный ресурс]: статья. – Режим доступа:<https://disqus.com/>
38. Bitbucket [Электронный ресурс]: статья. – Режим доступа:<https://bitbucket.org/product>
39. Nextdoor [Электронный ресурс]: статья. – Режим доступа:<https://nextdoor.com/>
40. "Usage Statistics of JavaScript as Client-side Programming Language on Websites". [Электронный ресурс]: статья. – Режим доступа:  
<https://w3techs.com/technologies/details/cp-javascript>
41. "ECMAScript 2020 Language Specification". Archived from the original on 8 May 2020
42. "Ajax". [Электронный ресурс]: статья. – Режим доступа:  
<https://www.xml.com/pub/a/2005/02/09/xml-http-request.html>
43. "Web-sockets standart". [Электронный ресурс]: статья. – Режим доступа: <https://websockets.spec.whatwg.org/>
44. "Usage statistics of JavaScript libraries for websites". *W3Techs*. Retrieved 9 April 2021.
45. "Using jQuery with Bootstrap". *clouddes.com*. 10 June 2019. Retrieved 17 March 2024.
46. "Vanilla JS". *vanilla-js.com*. 16 June 2020. Archived from the original on 16 June 2020. Retrieved 17 June 2020.
47. "Server-Side JavaScript Guide". *oracle.com*. Oracle Corporation. 11

48. "React – A JavaScript library for building user interfaces". *reactjs.org*. April 8, 2018.
49. "Chapter 1. What Is React? - What React Is and Why It Matters [Book]". *www.oreilly.com*. May 6, 2023.
50. Krill, Paul. "React: Making faster, smoother UIs for data-driven Web apps". May 15, 2014. *InfoWorld*. 2018-06-12.
51. Hemel, Zef. "Facebook's React JavaScript User Interfaces Library Receives Mixed Reviews", May 26, 2022.
52. Dawson, Chris. "JavaScript's History and How it Led To ReactJS". *The New Stack*. Aug 6, 2020..
53. Abramov, Dan; Tan, Lauren; Savona, Joseph; Markbåge, Sebastian. "Introducing Zero-Bundle-Size React Server Components", 21 December 2020.
54. "What is PostgreSQL?". *PostgreSQL 9.3.0 Documentation*. PostgreSQL Global Development Group. Retrieved September 20, 2013.
55. "Parquet and Postgres in the Data Lake | Crunchy Data Blog". *Crunchy Data*. May 3, 2022. Retrieved September 19, 2024.
56. "PostGIS". *postgis.net*. December 18, 2023. Retrieved December 18, 2023. PostGIS extends the capabilities of the PostgreSQL relational database by adding support for storing, indexing, and querying geospatial data.
57. "Temporal Extensions". *PostgreSQL Wiki*. December 18, 2023. Retrieved December 18, 2023. Postgres can be extended to become a Temporal Database. Such databases track the history of database content over time, automatically retaining said history and allowing it to be altered and queried.
58. "Orafce - Oracle's compatibility functions and packages". *GitHub.com*. December 17, 2023. Oracle RDBMS.



59. "pg\_dbms\_job". *GitHub.com*. November 8, 2023. PostgreSQL extension to schedules and manages jobs in a job queue similar to Oracle DBMS\_JOB package.
60. "PostgreSQL Clients". *wiki.postgresql.org*. October 18, 2023.