

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет радіофізики, електроніки та комп'ютерних систем
Кафедра комп'ютерної інженерії

**РОЗРОБКА ІНТЕРФЕЙСУ СИСТЕМИ АДМІНІСТРУВАННЯ
СПІВРОБІТНИКІВ ОРГАНІЗАЦІЇ**

Дипломна робота бакалавра
студента 4 року навчання
Спеціальність: 123 «Комп'ютерна інженерія»
Олександра ЛУКАШОВА

Науковий керівник
канд. техн. наук Віталій МАР'ЯНОВСЬКИЙ,
асистент кафедри комп'ютерної інженерії

Рецензент
кандидат фізико-математичних наук,
доцент Олексій НЕЧИПОРУК

До захисту допускаю:

Завідувач кафедрою Юрій БОЙКО

Ухвалено на засіданні кафедри “_____” _____ 2022 р., протокол № _____

ЗМІСТ

РЕФЕРАТ	3
СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	4
ВСТУП	5
1. ДОСЛІДЖЕННЯ ОБЛАСТІ ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	6
1.1. Дослідження системи адміністрування співробітників університету ..	6
1.2. Порядок надання доступу до інформації	6
1.3. Ідентифікація користувача	7
1.4. Технічна підтримка	7
1.5. Актуальність створення програмного забезпечення	8
1.6. Обґрунтування вибору мови програмування	8
2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	10
2.1. Діючий алгоритм роботи інтерфейсу програмного забезпечення	10
2.2. Технічні вимоги до розробки інтерфейсу адміністрування співробітників організації	11
2.3. Алгоритм роботи інтерфейсу програмного забезпечення АСО виконаний в рамках дипломного проекту.	13
3. МОДЕЛЬ СТВОРЕНОЇ БАЗИ ДАНИХ	15
4. РЕАЛІЗОВАНИЙ ІНТЕРФЕЙС СИСТЕМИ	22
4.1. Інтерфейс авторизації.....	22
4.2. Сторінка студента.....	28
4.3. Сторінка співробітника	43
4.4. Сторінка пошуку.....	64
4.5. Зміна паролю.....	72
4.6. Історія акаунту	75
4.7. Додавання користувачів до VPN груп.....	81
4.8. Відновлення паролю.....	84
ВИСНОВКИ	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	90

РЕФЕРАТ

Обсяг роботи за об'ємом складається з 92 сторінок, містить 57 рисунків, використано 23 інформаційні джерела.

Об'єктом дослідження є система адміністрування співробітників організації, її особливості, алгоритм роботи, та напрями її розбудови.

Станом на сьогодні ми маємо систему адміністрування співробітників організації Київського національного університету імені Тараса Григоровича Шевченка. Система працює на PHP 4 версії, яка на сьогодні застаріла, тому до неї неможливо додати нові модулі, що розширяють її функціонал.

Для усунення цього недоліку необхідно перейти на іншу сучасну технологію для модернізації системи.

Метою роботи є проведення дослідження, аналіз наявної системи та опис технологій, за допомогою яких буде можливість оновити програмне забезпечення до сучасного рівня.

У роботі описуються напрями побудови сучасної системи адміністрування співробітників організації.

Об'єкт дослідження має широке поле застосування в організаціях різних форм власності та має тенденцію збільшення попиту на розробку та впровадження систем адміністрування, таким чином аналіз та дослідження в цій сфері є необхідними.

Ключові слова: база даних, адміністрування, співробітник, технічна підтримка, python, Django, mysql, куки файли, запити, система.

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

Баги – помилки у програмі.

АСО – адміністрування співробітників організації.

ОС – операційна система.

КНУ - Київський національний університет імені Т.Г. Шевченко

Куки файли – тимчасові файли, які зберігаються на пристрої користувача

ВСТУП

КНУ є державним закладом вищої освіти IV рівня акредитації. Для забезпечення необхідного рівня якості освіти та функціонування інфраструктури, у складі університету передбачені окремі структурні підрозділи. Кожен підрозділ має свою область відповідальності та відповідний спектр задач. Одною з ключових задач управління закладом в усіх його підрозділах є АСО. Наявна система АСО застаріла, а внесення змін до програмного забезпечення є неефективним та затратним з огляду на використання застарілої версії мови програмування.

Проблематикою поточної версії системи є:

- a. Застаріла версія мови програмування
- b. Повільна швидкість роботи
- c. Вразливість у безпеці системи
- d. Проблема інтеграції із новими рішеннями
- e. Складність підтримки системи

Для усунення цього недоліку необхідно розробити нове програмне забезпечення АСО на основі сучасних версій мов програмування, що розширить функціонал системи, а також дозволить інтегруватись до вже існуючих систем університету.

Під час виконання дипломної роботи проведено дослідження та аналіз системи АСО КНУ, сформовані специфікації програмного забезпечення та розроблено новий інтерфейс системи АСО.

1. ДОСЛІДЖЕННЯ ОБЛАСТІ ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1. Дослідження системи адміністрування співробітників університету

На сьогодні КНУ має багато окремих структурних підрозділів. Кожен підрозділ має свою область відповідальності та відповідний спектр задач. Для виконання управлінських функцій адміністрації університету, та його окремих підрозділів потрібно мати актуальну та своєчасну інформацію по всім співробітникам установи. Задля вирішення цієї задачі була створена база даних адміністрування співробітників.

У 2002 році система отримала оновлення в вигляді новин університету.

Останні зміни проводилися у 2003 році – була впроваджена нова система реєстрації користувачів. З тих пір програмне забезпечення системи не оновлювалося.

Наявність системи для адміністрування співробітників університету актуально і на даний час. З огляду на використання застарілої версії мови програмування в адмініструванні наявної системи, необхідно провести модернізацію програмного забезпечення на основі сучасних версій мов програмування. У зв'язку з розробкою нового програмного забезпечення, ми отримуємо можливість змінювати, за необхідності, наявний функціонал і впровадити необхідні зміни з огляду на поточне законодавство та опираючись на потреби університету, його структурних підрозділів, та досвід роботи спеціалістів які працювали з системою.

1.2. Порядок надання доступу до інформації

У КНУ є багато підрозділів, працює багато співробітників, яким необхідно мати доступ до інформації. Доступ до інформації повинні мати лише ті співробітники, яким це право буде делеговано адміністрацією

університету. Кожен співробітник повинен бути обмежений в доступах до системи згідно делегованих прав.

Співробітники які мають доступ системи повинні мати можливість читання, редагування, додавання, вивід інформації на зовнішні пристрої і ресурси, та видалення даних в усій базі, або її окремій області в залежності від наявних дозволів.

1.3. Ідентифікація користувача

Для того щоб переглянути чи надати інформацію співробітнику необхідно зайти на форму програмного застосунку. Після переходу на сторінку авторизації йому необхідно пройти ідентифікацію. Воно відбувається наступним чином: працівник вводить логін та пароль, ці дані передаються на сервер, де порівнюються з таблицею користувачів. Якщо логін та пароль введено правильно сервер, перевіряє термін дії акаунту та надає право даному користувачу отримати доступ до області даних та функціоналу системи співробітників згідно наявних дозволів. Після цього це рішення передається до форми клієнта і дає змогу увійти до системи співробітників та зробити необхідні операції. В іншому випадку система дає відповідь, що користувачу недостатньо прав для доступу до системи співробітників.

1.4. Технічна підтримка

Для стабільного функціонування програмного забезпечення необхідно постійне функціонування служби технічної підтримки, та адміністрування баз даних користувачів та даних співробітників яка своєчасно зможе вирішити проблему доступу до бази даних співробітників, а в разі необхідності вирішити проблеми на апаратному або програмному рівні.

Для стабільного функціонування програмному забезпеченню необхідно впровадити систему своєчасних оновлень.

Для інформування про необхідність зміни якихось функцій або усунення багів нам необхідно розробити форму зворотнього зв'язку з адміністратором

щоб користувач якщо знайшов якусь проблему у роботі програми міг її описати та направити на вирішення.

1.5. Актуальність створення програмного забезпечення

Одною з ключових задач управління закладом в усіх його підрозділах є управління співробітниками організації. Діюча система АСО застаріла, а внесення змін до програмного забезпечення є неефективним з огляду на використання застарілої версії мови програмування.

У зв'язку з цим необхідно розробити та впровадити нове програмне забезпечення у тому числі з можливістю розширення функціоналу.

1.6. Обґрунтування вибору мови програмування

Аналізуючи минулу систему, ми можемо побачити, що всі сервіси, до яких вона посилалась, працюють на ОС Linux. Сучасні подібні системи також працюють в більшості на ОС Linux. Отже, нам необхідно обрати таку технологію, яка була майже ідеально сумісна з ОС Linux.

Користуючись відомостями про найбільш сумісні мови та технології, ми прийшли до висновку, що на ОС Linux будуть достатньо працювати наступні мови: C, C++, Python, Bash.

На чистому C буде неможливо написати достатньо функціональний веб застосунок.

У C++ наявно більше можливостей порівняно з C, але й він не налаштований для роботи з веб застосунками.

Python гармонічно співпрацює з ОС Linux, бо є частиною Unix подібних систем. Має багато фреймворків за допомогою яких, ми можемо створити веб-сторінки та розробити необхідну систему з підключенням бази даних і з легкістю запустити на Unix подібних системах.

Bash використовуються для автоматизації рішень у Unix подібних системах.

Отже, ми обираємо Python, тепер оберемо фреймворк. У ході аналізу, двома претендентами залишилися Flask та Django. Django більш модульний елемент розробки сайтів, має свою панель адміністрування та можливості підключення до бази даних. В той час як Flask потребує розробки всіх модулів самостійно. Тому було вирішено використовувати саме Django для розробки необхідного програмного забезпечення.

2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Діючий алгоритм роботи інтерфейсу програмного забезпечення

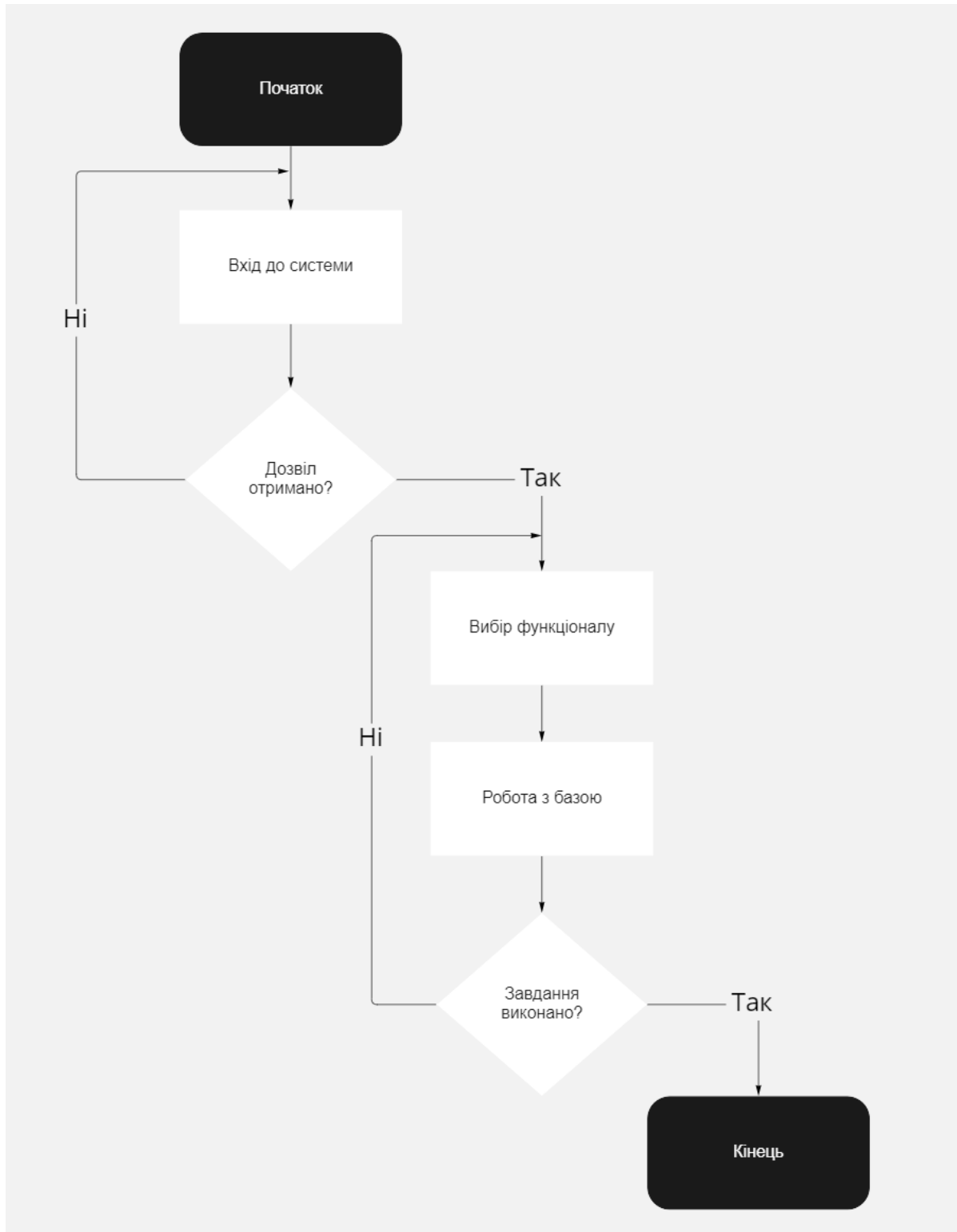


Рис. 1 – Схема наявного алгоритму роботи інтерфейсу програмного забезпечення АСО.

- надсилання запиту користувачем
- перевірка чи дозволено користувачу вхід до системи
- перевірка прав доступу на отримання інформації
- перевірка прав на редагування інформації
- відправка результату до користувача
- вхід у програму або відмова
- робота з базою даних у відповідності до функціоналу програмного забезпечення та виданих прав
- зберігання змін
- робота завершена [так ні]
- вихід

2.2. Технічні вимоги до розробки інтерфейсу адміністрування співробітників організації

Надані науковим керівником вимоги

2.2.1. Програмні вимоги

2.2.1.1. Розробити систему адміністрування користувачів на основі поточної:

- Створення користувачів
- Редагування користувачів
- Пошук користувачів
- Логування змін у системі

2.2.1.2. Додати нові функції:

- Керування користувачами у VPN групах
- Створення функціоналу для користувача:
 - Зміна профілю
 - Пошук користувачів

➤ Зміна пароля, пошти

2.2.1.3. Відновлення пароля за допомогою електронної пошти

- Механізм розмежування прав користувачів
- Обмеження часу життя облікового запису

2.2.2. Системні вимоги:

- Мова Python v3.9.7 та вище
- Фреймворк Django 4.0.4 та вище
- База даних MySQL v2.0.2 та вище

Додаткові вимоги:

Неможна вносити зміни в структуру базу даних

Узгоджені в процесі розробки програмного забезпечення вимоги

1. Через обмежень фреймворку Django, було необхідно внести зміни до структури бази даних.
2. Використання в процесі розробки:
 - Бібліотека PyMySQL v1.0.2 та вище
 - Бібліотека charset-normalizer v2.0.12
 - Бібліотека email-validate v1.1.2 та вище
 - Бібліотека requests v2.27.1 та вище

Наразі для роботи розробленого програмного забезпечення потрібно виконання наступних апаратних вимог:

2.7. Апаратні вимоги:

2.7.1. Мінімальні:

- Unix подібна система (Ubuntu, Debian, Red Hat, CentOS)
- Процесор: x86_64 або новіший (Intel не нижче Pentium 4, AMD не нижче Athlon 64)

- ОЗУ: 256 Мб
- Дисковий простір: 7,5 ГБ
- Простір підкачування: 256Мб

2.8 Рекомендовані:

- Unix подібна система (Ubuntu, Debian, Red Hat, CentOS)
- Процесор: Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz
- ОЗУ: 1Гб
- Дисковий простір: 15Гб
- Простір підкачування: 256Мб

2.3. Алгоритм роботи інтерфейсу програмного забезпечення АСО виконаний в рамках дипломного проекту.

- вхід на початкову сторінку інтерфейсу АСО
- вибір чи необхідно відновити пароль
- за необхідністю зміна паролю за допомогою поштового сервісу
- перевірка логіну, паролю, рівня доступу, терміну дії аккаунта
- видання користувачу дозволеного функціоналу згідно з рівнем доступу
- формування запиту
- перевірка чи дозволена така дія даному користувачу
- робота з обраною дією
- робота завершена [так ні]
- вихід у головне меню чи кінець роботи

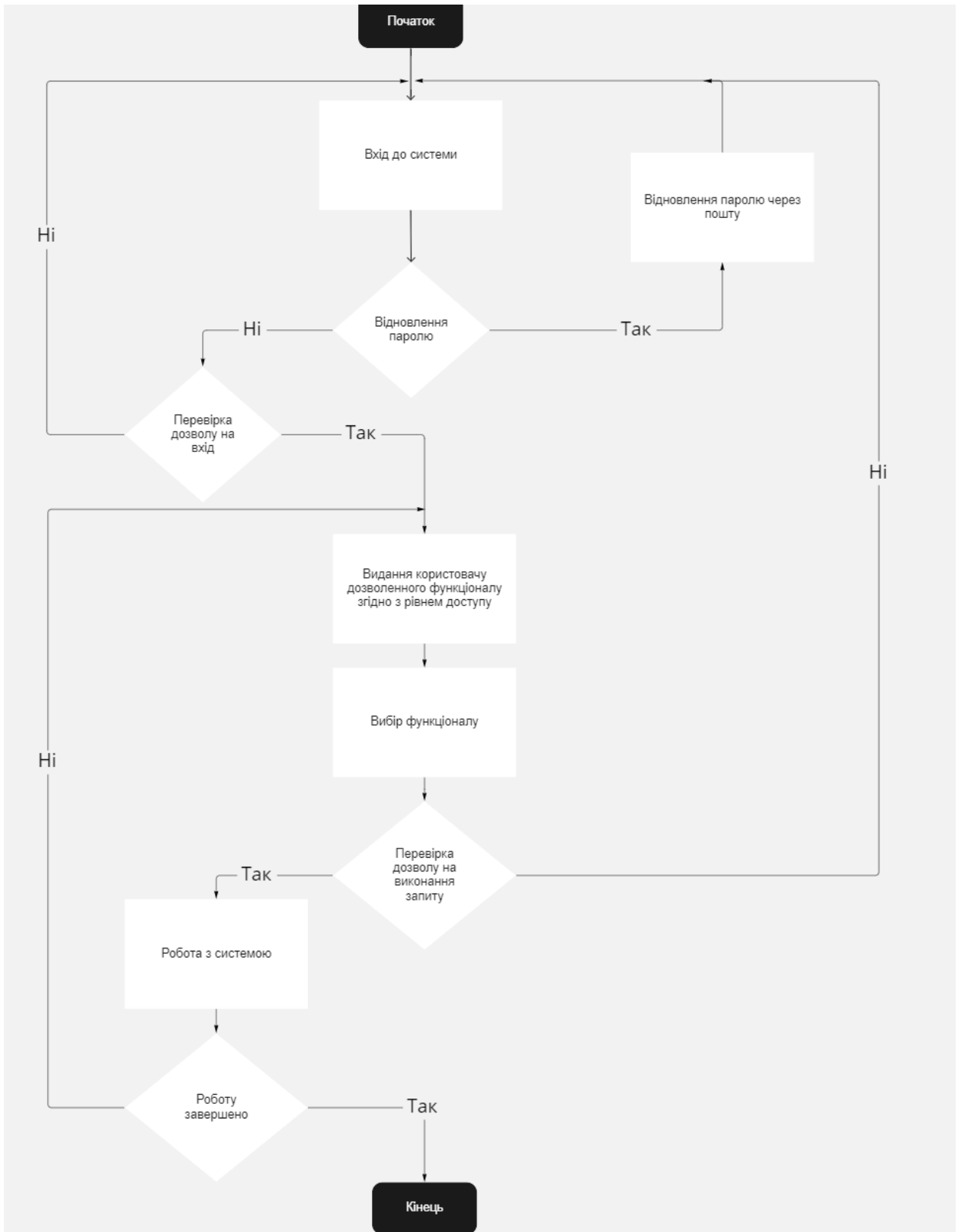


Рис. 2 – Схема узгодженого алгоритму роботи інтерфейсу програмного забезпечення АСО.

3. МОДЕЛЬ СТВОРЕННОЇ БАЗИ ДАНИХ

Таблиця `access_group`

```
class AccessGroup(models.Model):
    group_id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=50, blank=True,
null=True)
    description = models.CharField(max_length=255,
blank=True, null=True)
```

Таблиця `access_group` відповідає за доступ до ресурсів за допомогою VPN.

Поле `group_id` – цілочисельний показник групи користувача, який є первинним ключем для таблиці.

Поле `name` – рядок, максимальною довжиною в 50 символів, що відповідає ім'я групи VPN.

Поле `description` – рядок, максимальної довжини у 255 символів, що відповідає за опис групи.

Таблиця `dept_all`

```
class DeptAll(models.Model):
    dept_id = models.AutoField(primary_key=True)
    dept_ename = models.CharField(unique=True,
max_length=64)
    dept_uname = models.CharField(unique=True,
max_length=64)
    copymail = models.CharField(max_length=50)
```

Таблиця `dept_all` відповідає за підрозділи університету

Поле `dept_id` – автоматичне цілочисельне число, що збільшується на 1 кожного разу при додаванні нового запису, в таблиці є первинним ключем та відповідає за ідентифікатор підрозділу.

Поле `dept-ename` – рядок, що має максимальну довжину в 64 символи, відповідає за назву підрозділу англійською мовою.

Поле `dept-uname` – рядок, що має максимальну довжину в 64 символи, відповідає за назву підрозділу українською мовою.

Поле `soremail` – рядок, що має максимальну довжину в 50 символів, відповідає за імейл, за яким можливо звернутись до відповідальної людини за визначений підрозділ.

Таблиця `faculty_all`

```
class FacultyAll(models.Model):
    faculty_id =
models.PositiveIntegerField(primary_key=True)
    faculty = models.CharField(max_length=30, blank=True,
null=True)
    dept_id = models.IntegerField()
```

Таблиця `faculty_all` відповідає за факультети університету.

Поле `faculty_id` – позитивне цілочисельне значення, що являє собою первинний ключ та відповідає за унікальний номер факультету.

Поле `faculty` – рядок, максимальна довжина якого 30 символів, що відповідає за назву факультету.

Поле `dept_id` – цілочисельне значення, що відповідає за номер підрозділу, до якого належить факультет.

Таблиця `title_all`

```
class TitleAll(models.Model):
    title_id = models.AutoField(primary_key=True)
    title_uname = models.CharField(unique=True,
max_length=30)
    title_ename = models.CharField(unique=True,
max_length=30)
```

Таблиця `title_all` відповідає за посаду людини в системі

Поле `title_id` – автоматичне цілочисельне число, що збільшується на 1 кожного разу при додаванні нового запису, в таблиці є первинним ключем та відповідає за унікальний ідентифікатор посади.

Поле `title.uname` – унікальний рядок, максимальної довжини в 30 символів, що відповідає за назву посади українською мовою

Поле `title.ename` – унікальний рядок, максимальної довжини в 30 символів, що відповідає за назву посади англійською мовою

Таблиця `users_access_group`

```
class UsersAccessGroup(models.Model):
    user = models.OneToOneField('UsersAll',
models.DO_NOTHING, primary_key=True)
    group = models.ForeignKey(AccessGroup,
models.DO_NOTHING)
    desc = models.TextField(blank=True, null=True)
```

Таблиця `users_access_group` відповідає за з'єднання людини та групи доступів за VPN.

Поле `user` – поле з відношенням один до одного, що посилається на первинний ключ у таблиці `UserAll` та служить ідентифікатором користувача

Поле `group` – поле з зовнішнім ключем, що посилається на первинний ключ таблиці `AccessGroup` та являє собою ідентифікатор групи доступу до VPN.

Поле `desc` – текстове поле, що може бути незаповненим, позначає причину, на основі якої видано право користування групою доступу

Таблиця `users_history`

```
class UsersHistory(models.Model):
    id = models.BigAutoField(primary_key=True)
    user_id = models.PositiveIntegerField()
    date = models.DateTimeField()
    action = models.CharField(max_length=10)
    creator = models.CharField(max_length=16)
    reason = models.TextField(blank=True, null=True)
    ip = models.CharField(max_length=15, blank=True,
null=True)
```

Таблиця `users_history` відповідає за логування дій користувача

Поле `id` – було додане до таблиці, оскільки Django не вміє працювати з таблицями без первинного ключа, це велике автоматичне поле, що позначає унікальний номер запису в таблиці та являє собою первинний ключ.

Поле `user_id` – додатній цілочисельний номер, що відповідає за унікальний ідентифікатор користувача.

Поле `date` – поле дати та часу, що записує дату та час зміни інформації

Поле `action`– рядок довжиною у 10 символів, що записує дію, при якій відбулись зміни.

Поле `creator`– рядок, довжиною у 16 символів, що записує користувача, що провів операцію.

Поле `reason`– текстове поле, що вказує, що саме змінилося, але є можливістю залишити це поле порожнім.

Поле `ip`– рядок довжиною у 15 символів, що зберігає ip-адресу з якою було проведено операцію.

Таблиця `users_all`

```
class UsersAll(models.Model):
    user_id = models.AutoField(primary_key=True)
    dept_id = models.PositiveIntegerField()
    state = models.CharField(max_length=6)
    login = models.CharField(unique=True, max_length=16)
    passwd = models.CharField(max_length=40)
    fname = models.CharField(max_length=50)
    lname = models.CharField(max_length=50)
    mname = models.CharField(max_length=50)
    contr_quest = models.CharField(max_length=255)
    contr_answ = models.CharField(max_length=255)
    title = models.CharField(max_length=30)
    email = models.CharField(max_length=50)
    phone = models.CharField(max_length=20, blank=True,
null=True)
    exp_date = models.DateTimeField(blank=True, null=True)
    welcome_msg = models.CharField(max_length=255,
blank=True, null=True)
    admin = models.CharField(max_length=1)
    document_id = models.CharField(max_length=16)
    creator = models.CharField(max_length=16)
    operator = models.CharField(max_length=1)
    mgroup_id = models.IntegerField()
    title_id = models.PositiveIntegerField()
    admin_level = models.PositiveIntegerField()
```

Таблиця `users_all` відповідає за всю інформацію про користувача

Поле `user_id` – автоматичне цілочисельне число, що збільшується на 1 кожного разу при додаванні нового запису, в таблиці є первинним ключем та відповідає за унікальний номер користувача

Поле dept_id – додатній цілочисельний номер, що відповідає за ідентифікатор підрозділу, до якого належить користувач

Поле state – рядок довжиною до 6 символів, позначає стан аккаунту

Поле login– унікальний рядок довжиною до 16 символів, відповідає за логін користувача

Поле passwd– рядок довжиною до 40 символів, відповідає за пароль користувача

Поле fname– рядок довжиною до 50 символів, відповідає за ім'я користувача

Поле lname– рядок довжиною до 50 символів, відповідає за прізвище користувача

Поле mname– рядок довжиною до 50 символів, відповідає за ім'я по батькові користувача

Поле contr_quest– рядок довжиною до 255 символів, відповідає за контрольне питання користувача

Поле contr_answ– рядок довжиною до 255 символів, відповідає за контрольну відповідь користувача

Поле title– рядок довжиною до 30 символів, відповідає за опис посади, яку займає користувач

Поле email– рядок довжиною до 50 символів, відповідає за імейл користувача

Поле phone– рядок довжиною до 20 символів, відповідає за телефон користувача, це поле не обов'язкове до заповнення, тому може бути пустим.

Поле exp_date– поле дати та часу, відповідає за дату закриття аккаунту користувача

Поле welcome_msg– рядок довжиною до 255 символів, відповідає за вітальне повідомлення користувача, може бути пустим.

Поле admin– рядок довжиною в 1 символ, відповідає за те, чи є цей користувач адміністратором.

Поле `document_id`– рядок довжиною до 16 символів, відповідає за номер посвідчення.

Поле `creator`– рядок довжиною до 16 символів, відповідає за людину, що створила цього користувача в системі

Поле `operator`– рядок довжиною в 1 символ, відповідає за те, чи є цей користувач оператором

Поле `mgroup_id` – цілочисельний номер.

Поле `title_id` – додатній цілочисельний номер ідентифікатору посади користувача

Поле `admin_level` – додатній цілочисельний номер рівня адміністрування користувача.

Таблиця RestorePassword

```
class RestorePassword(models.Model):
    record_id = models.AutoField(primary_key=True)
    user = models.PositiveIntegerField()
    exp_date = models.DateTimeField()
    created_date = models.DateTimeField()
    ip = models.CharField(max_length=15, blank=True,
null=True)
    temporary_pass = models.CharField(max_length=40)
```

Таблиця `RestorePassword` була додана для реалізації механізму відновлення паролю користувача.

Поле `record_id` – автоматичне цілочисельне число, що збільшується на 1 кожного разу при створенні нового запису, відповідає за унікальний номер запису та являє собою первинний ключ для цієї таблиці

Поле `user` – додатній цілочисельний номер, що позначає користувача з таблиці `UserAll`.

Поле `exp_date` – поле дати та часу, яке записує до якого часу буде дійсна форма зміна паролю.

Поле `created_date` – поле дати та часу, яке записує час створення заявки на відновлення паролю

Поле `ip` – рядок довжиною до 15 символів, що записує ір-адресу з якої була заявка на відновлення паролю.

Поле `temporary_pass` – рядок довжиною до 40 символів, що записує тимчасове значення за допомогою якого є можливість змінити пароль.

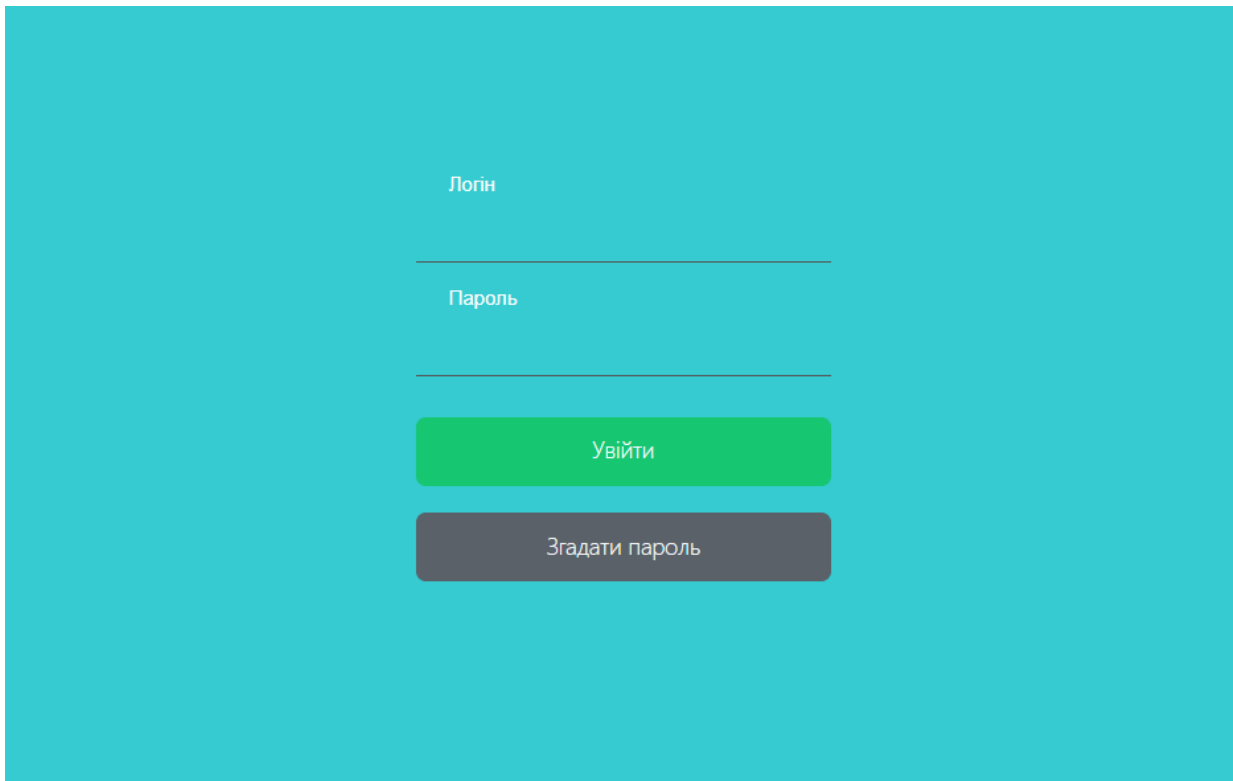
4. РЕАЛІЗОВАНИЙ ІНТЕРФЕЙС СИСТЕМИ

4.1. Інтерфейс авторизації

Коли користувач заходить до системи, він потрапляє на початкову сторінку входу до web панелі (Рис. 3), в нього перевіряються куки файли. Якщо є куки логіну та статусу входу, користувач одразу перенаправляється до початкової сторінки системи в залежності від його рівня адміністратора. Якщо куки файли відсутні - запитується логін та пароль. Після заповнення полів, система починає шукати в базі логін, якщо він знайдений, то далі береться пароль, хешується створеним методом MD5, та перевіряється із значенням, що знаходиться у базі.

```
def hashMDPass(password):  
    hash_object = hashlib.md5(password.encode())  
    md5_hash = hash_object.hexdigest()  
    return md5_hash
```

Потім перевіряє, чи термін дії аккаунта сплив. Якщо ні, перевіряється рівень адміністрування користувача, якщо він більше або рівний 1, то створюється куки, що зберігає логін користувача та статус входу(вхід здійснений чи ні) та рівень адміністратора. Після цього користувач отримує доступ та перенаправляється на початкову сторінку системи (Рис. 8)



Логін

Пароль

Увійти

Згадати пароль

Рис. 3 – Початкова сторінка входу до web панелі

В ході роботи авторизації були передбачені наступні помилки:

- Неправильний логін або пароль: Помилка: «Неправильний логін/пароль.» (Рис. 4);
- Якщо і логін, і пароль ведені правильно, перевіряється строк дії аккаунти. Якщо він сплив, виводиться помилка: Помилка «Термін дії аккаунту сплив. Зверніться до адміністратора.» (Рис. 5);
- Якщо і логін, і пароль ведені правильно, строк дії акаунти актуальний, то перевіряється рівень адміністратора, якщо рівня адміністратора недостатньо, виводиться помилка: Помилка «У вас недостатньо прав на вхід» (Рис. 6).

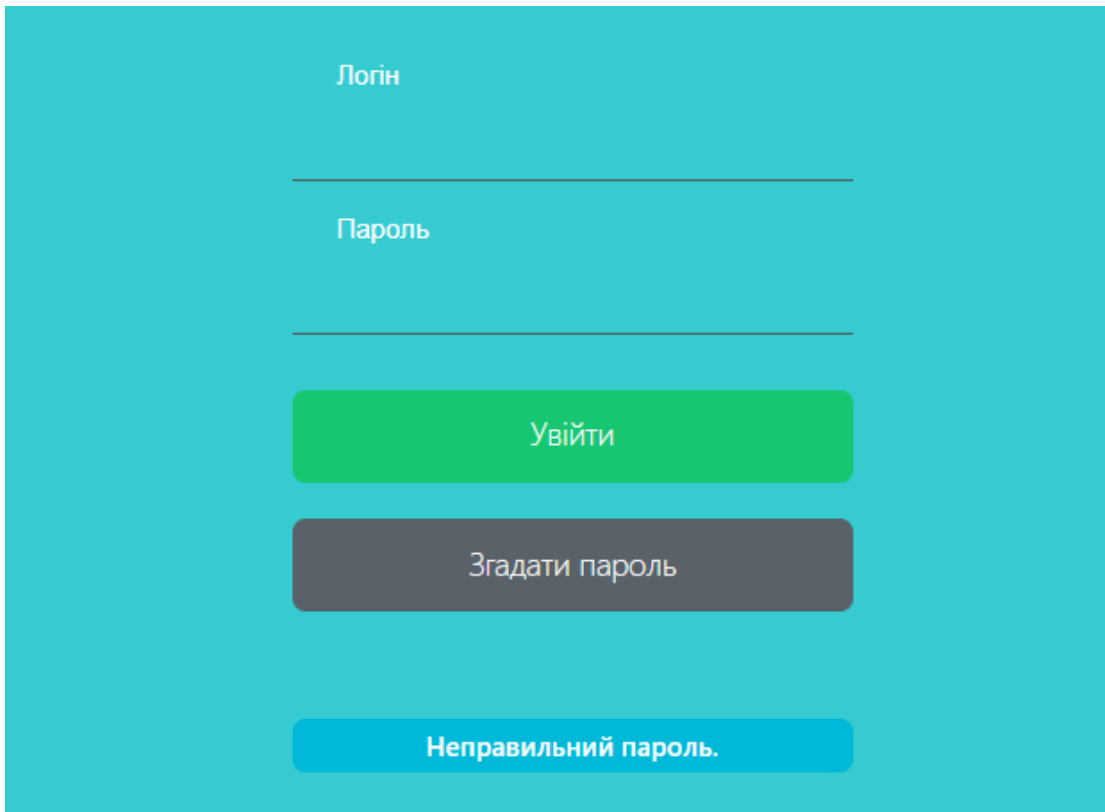


Рис. 4 – Помилка: Неправильний логін/пароль.

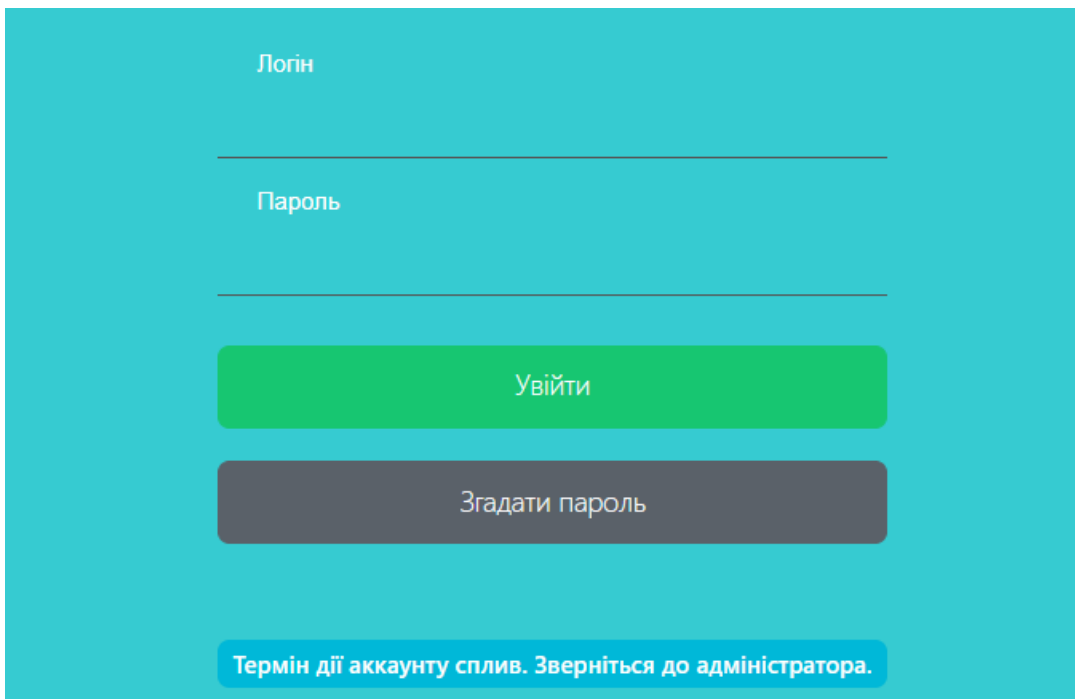
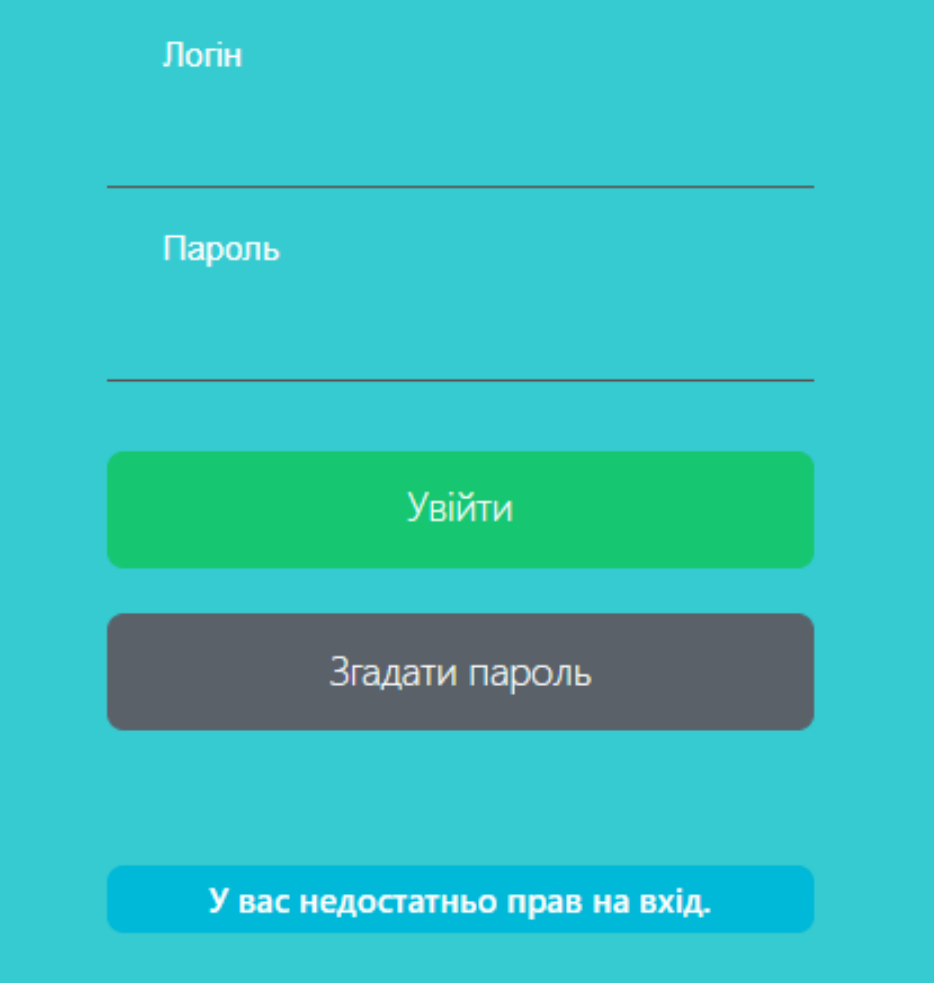


Рис. 5 – Помилка: Термін дії акаунту сплив. Зверніться до адміністратора.



Логін

Пароль

Увійти

Згадати пароль

У вас недостатньо прав на вхід.

Рис. 6 – Помилка: У вас недостатньо прав на вхід

На всіх сторінках перевіряються куки файли. Якщо дані про користувача, записані у куках є - користувач переадресується на початкову сторінку, в залежності від рівня адміністратора.

Фрагмент коду авторизації користувача.

```

def login(request):
    if not check_cookies(request.COOKIES.get('username'),
request.COOKIES.get('login_status')):
        if 'timePass' in request.COOKIES:
            info = "Перевірте тимчасовий пароль на пошті"
        else:
            info = ""
    if request.method == "GET":
        form = userAllForm()
        data = {
            'form': form,
            'info': info
        }
        response = render(request, 'main/login.html',
data)

        return response

    if request.method == "POST":
        form = userAllForm(request.POST)
        login = form.data["login"]
        password = form.data["passwd"]
        users =
UsersAll.search_all(UsersAll.objects.filter(login=form.data["log
in"]))

        logD = UsersAll.objects.all()
        for el in logD:
            if login == el.login:
                if request.COOKIES.get('timePass') ==
None:
                    timepass = ""
                else:
                    timepass =
request.COOKIES.get('timePass')
                    if hashMDPass(password) == el.passwd or
check_password(password, timepass):
                        if el.exp_date.timestamp() >
datetime.now().timestamp():
                            data = {
                                'lvl': el.admin_level
                            }
                            if el.admin_level >= 2:
                                return
create_cookies(request, el.admin_level, login,
'main/index.html', data)
                            elif el.admin_level == 1:
                                return
create_cookies(request, el.admin_level, login,
'main/user_home.html', data)
                            else:

```

```

        info = "У вас недостатньо
прав на вхід."
        break
    else:
        info = "Термін дії аккаунту
сплив. Зверніться до адміністратора."
        break
    else:
        info = "Неправильний пароль."
        break
    else:
        info = "Такого логіну не існує"
form = userAllForm()
data = {
    'form': form,
    'info': info
}
response = render(request, 'main/login.html', data)
return response
else:
    if int(request.COOKIES.get('iqlvl')) >= 2:
        response = redirect('home')
        return response
    elif int(request.COOKIES.get('iqlvl')) < 2:
        response = redirect('user_home')
        return response
        return response

```

4.2. Сторінка студента

З початку користувач отримує доступ до початкової сторінки. (Рис. 7) Якщо, користувач з правами адміністратора рівень 1 – то користувач зможе передивитись свою сторінку (Рис. 8) (Рис. 9) з інформацією про себе, змінити привітальне повідомлення, контрольне питання та відповідь, імейл та телефон. Якщо користувач змінює імейл – то додатково перевіряється чи є цей імейл дійсним. Також у системі користувач може знайти інформацію про інших співробітників організації, а саме ім'я, прізвище, ім'я по батькові, факультет та імейл аналогічно до інформації про самого користувача. При натисканні кнопки редагувати, відкривається вікно редагування інформації про користувача. (Рис. 10) Сторінка пошуку користувачів у системі (Рис. 11), максимальна кількість виведених записів 5, інакше буде виведено запис – «Уточніть запит».

На пошуку користувачів у системі були оброблені наступні запити:

- При натисканні пошуку без заповненої форми – виводиться нагадування, про те, що ви не заповнили нічого з форми: Помилка «Уведіть хоча б один із пунктів для пошуку» (Рис. 12)

- Якщо користувача не знайдено, виводиться напис: Помилка «За такими даними користувача не знайдено.» (Рис. 13)

Якщо користувач існує, знизу виводиться інформації про нього (Рис. 14)

Останній пункт інтерфейсу користувача – зміна паролю (Рис. 15), де він може задати власний пароль.

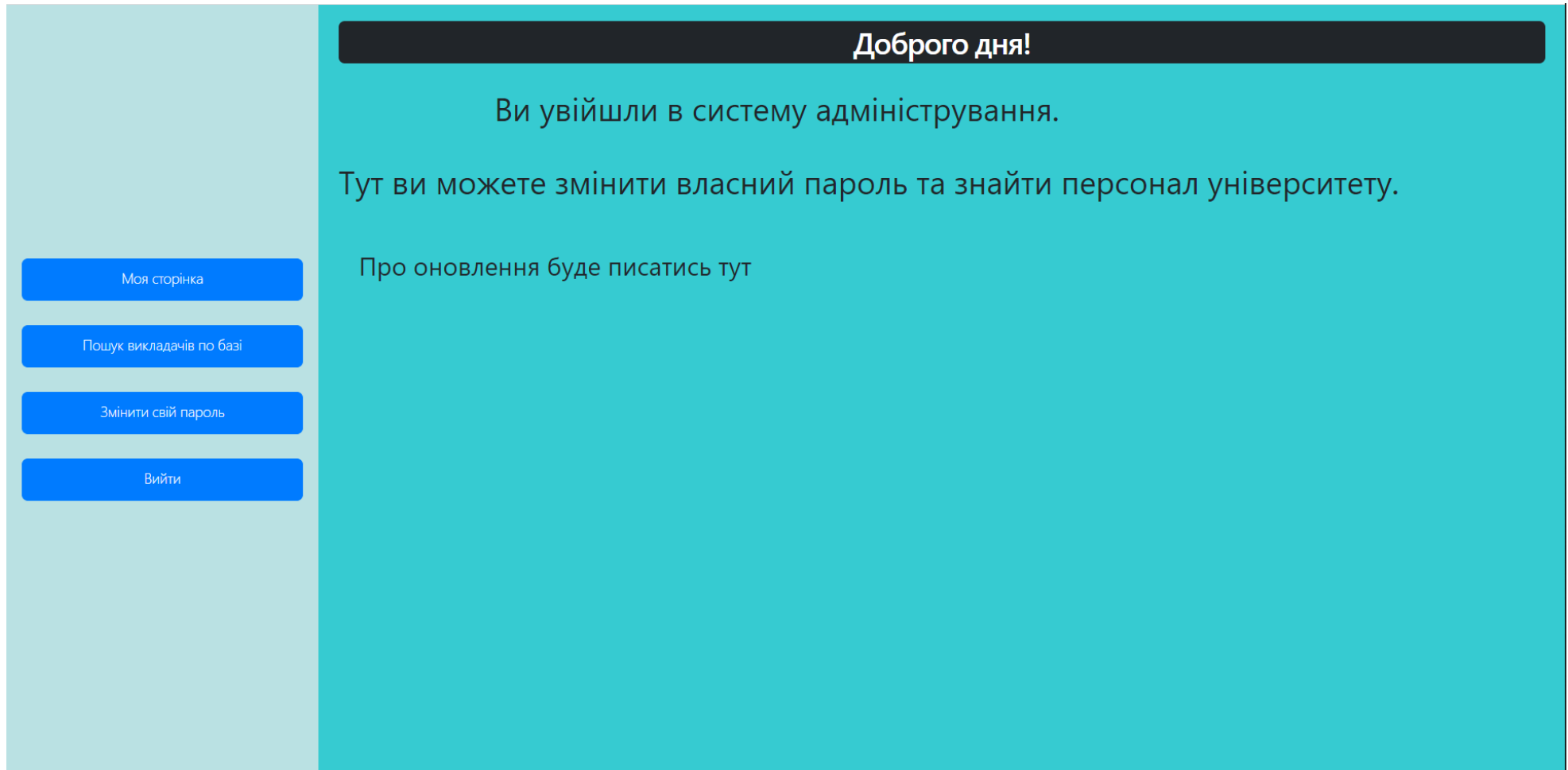


Рис. 7 – Початкова сторінка.

Департамент : The Faculty of Biology

Стан : OPEN

Логін : feq

Ім'я : nvjodnkj

Прізвище : vdnwkj

По батькові : j bfkj

Контрольне питання : -

Контрольна відповідь : -

Опис посади : -

Моя сторінка

Пошук викладачів по базі

Змінити свій пароль

Вийти

Рис. 8 – Сторінка інформації про користувача (частина 1)

The image shows a user profile page with a light blue sidebar on the left and a dark grey main content area on the right. The sidebar contains four blue buttons: 'Моя сторінка', 'Пошук викладачів по базі', 'Змінити свій пароль', and 'Вийти'. The main content area displays user information in a list of horizontal bars, each with a cyan border. The information includes: 'Привітальне повідомлення : -', 'Адміністратор : n', 'Ваш номер залікової книги : 31231', 'Створювач : bloodelf', 'Оператор : n', 'Мгрууп : 1', 'Посада : undefined', and 'Рівень адміністратора : 1'. At the bottom of the main content area, there is a black button labeled 'Редагувати'.

Моя сторінка

Пошук викладачів по базі

Змінити свій пароль

Вийти

Привітальне повідомлення : -

Адміністратор : n

Ваш номер залікової книги : 31231

Створювач : bloodelf

Оператор : n

Мгрууп : 1

Посада : undefined

Рівень адміністратора : 1

Редагувати

Рис. 9 – Сторінка інформації про користувача (частина 2)

Моя сторінка

Пошук викладачів по базі

Змінити свій пароль

Вийти

Контрольне питання : -

Контрольна відповідь : -

Опис посади : -

Імейл : blodelf.alex@gmail.com

Телефон : -

Дата кінця аккаунту : Dec. 9, 2027, 12:43 p.m.

Привітальне повідомлення : -

Адміністратор : n

Ваш номер залікової книги : 31231

Рис. 10 – Вікно редагування інформації про користувача

Сторінка пошуку персоналу

Будь ласка, заповніть хоча б одне з полів

Прізвище

Ім'я

Побатькові

Email

Пошук

Моя сторінка

Пошук викладачів по базі

Змінити свій пароль

Вийти

Рис. 11 – Сторінка пошуку користувачів у системі

Будь ласка, заповніть хоча б одне з полів

Прізвище

Ім'я

Побатькові

Email

Пошук

Уведіть хоча б один із пунктів для пошуку

Рис. 12 – Помилка: Уведіть хоча б один із пунктів для пошуку

Будь ласка, заповніть хоча б одне з полів

Прізвище
hdwfh

Ім'я

Побатькові

Email

Пошук

За такими даними користувача не знайдено.

The image shows a search interface on a teal background. At the top, a dark grey banner contains the text 'Будь ласка, заповніть хоча б одне з полів'. Below this are four input fields: 'Прізвище' (with the value 'hdwfh'), 'Ім'я', 'Побатькові', and 'Email'. A green 'Пошук' button is centered below the fields. At the bottom, a yellow error message box states 'За такими даними користувача не знайдено.'

Рис. 13 – Помилка: За такими даними користувача не знайдено.

Прізвище

Ім'я

kdnov

Побатькові

Email

Пошук

Прізвище	Ім'я	По батькові	Факультет	Імейл
kjvfwo	kdnov	kvndkl	The Faculty of Chemistry	talabor548@nzaif.com

Рис. 14 – Інформація про користувача згідно з запитом.

Сторінка зміни паролю

Новий пароль

Підтвердження паролю

Змінити

Моя сторінка

Пошук викладачів по базі

Змінити свій пароль

Вийти

Рис. 15 – Зміна паролю

В ході зміни паролю передбачені наступні помилки:

- Обов'язкове заповнення поля паролю та підтвердження: «Заповніть це поле» (Рис. 16)
- Перевірка на довжину паролю: «Мінімальна довжина паролю 8 символів.» (Рис. 17)
- Перевірка на співпадіння паролю: «Паролі не співпадають.» (Рис. 18)

При правильному введенні даних виводиться повідомлення про успішну зміну паролю: Пароль успішно змінено. (Рис. 19)

Сторінка зміни паролю

Новий пароль

Підтвердження паролю

Змінити

Моя сторінка

Пошук викладачів по базі

Змінити свій пароль

Вийти

Новий пароль

Підтвердження паролю

Змінити

Заполните это поле.

The image shows a web interface for changing a password. On the left is a vertical sidebar with four blue buttons: 'Моя сторінка', 'Пошук викладачів по базі', 'Змінити свій пароль', and 'Вийти'. The main content area has a teal header with the title 'Сторінка зміни паролю'. Below the header, there are two input fields: 'Новий пароль' and 'Підтвердження паролю'. The 'Новий пароль' field is currently empty and has a red border, with a white tooltip containing an exclamation mark icon and the text 'Заполните это поле.' (Fill in this field). Below the input fields is a green button labeled 'Змінити'.

Рис. 16 – Помилка: «Заповніть це поле»

The image shows a web interface for changing a password. On the left is a vertical sidebar with four blue buttons: "Моя сторінка", "Пошук викладачів по базі", "Змінити свій пароль", and "Вийти". The main content area has a title "Сторінка зміни паролю" in a light grey header. Below the title are two input fields: "Новий пароль" and "Підтвердження паролю". A red error message "Мінімальна довжина паролю 8 символів." is displayed in a red bar below the second field. A green "Змінити" button is at the bottom.

Сторінка зміни паролю

Новий пароль

Підтвердження паролю

Мінімальна довжина паролю 8 символів.

Змінити

Моя сторінка

Пошук викладачів по базі

Змінити свій пароль

Вийти

Рис. 17 – Помилка: «Мінімальна довжина паролю 8 символів.»

The image shows a web interface for changing a password. On the left, there is a vertical sidebar with four blue buttons: "Моя сторінка", "Пошук викладачів по базі", "Змінити свій пароль", and "Вийти". The main content area has a dark grey header with two input fields: "Пароль" and "Підтвердження паролю". Below these fields, a red error message bar displays the text "Паролі не співпадають". A green "Змінити" button is positioned below the error message.

Рис. 18 – Помилка: «Паролі не співпадають.»

Пароль

Пароль

Підтвердження паролю

Підтвердження паролю

Моя сторінка

Пошук викладачів по базі

Змінити свій пароль

Вийти

Пароль успішно змінено.

Змінити

Рис. 19 – Пароль успішно змінено.

4.3. Сторінка співробітника

Якщо рівень адміністрування користувача більше за 1, виводиться наступне меню: Ви увійшли в систему адміністрування Інтернет доступу користувачів (Рис. 20)

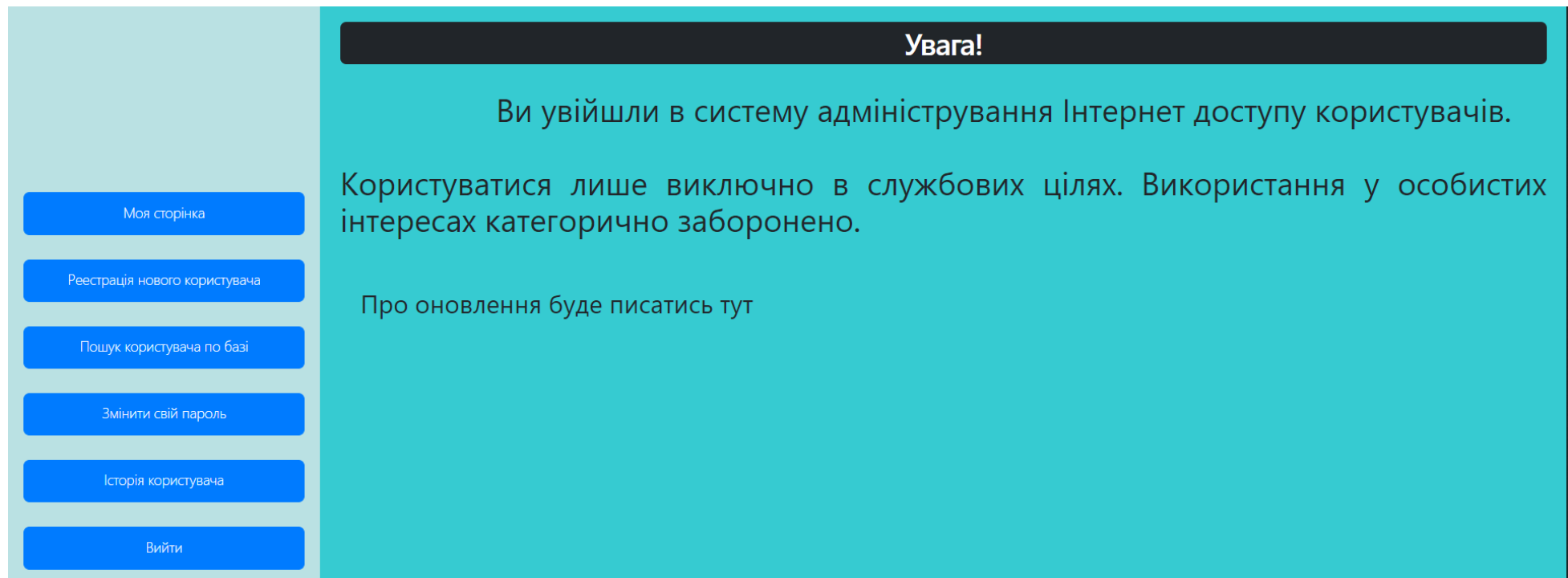


Рис. 20 – Ви увійшли в систему адміністрування Інтернет доступу користувачів

Сторінка інформації про користувача з рівнем доступу 2 виглядає так само як у користувача з рівнем доступу 1. Зміни настають, коли користувач хоче змінити інформацію про себе або співробітників свого підрозділу (Рис. 21-22) Користувач може змінити будь-яку інформацію окрім логіну.

Користувач може передивитись сторінку інформації про себе, зареєструвати нового користувача, знайти студента або співробітника в системі, змінити свій пароль та подивитись зміни за логіном користувача.

Були оброблені наступні помилки:

- Імейл не дійсний. (Рис. 23)
- Ви не можете змінити рівень адміністратору на вищий за свій. (Рис. 24)

Департамент : The Faculty of Geography

Стан : OPEN

Логін : Gihiruct

Ім'я : Аркакий

Прізвище : Леонтев

По батькові : Васильевич

Контрольне питання : -

Контрольна відповідь : -

Опис посади : -

Моя сторінка

Реєстрація нового користувача

Пошук користувача по базі

Змінити свій пароль

Історія користувача

Вийти

Рис. 21 – Сторінка редагування інформації користувача з рівнем доступу 2 про себе, та співробітників свого підрозділу.

Дата кінця аккаунту : Dec. 11, 2027, midnight

Привітальне повідомлення : -

Адміністратор : n

Ваш номер залікової книги : 1234

Створювач : bloodelf

Оператор : n

Мгрууп : 1

Посада : cadet

Рівень адміністратора : 2

Зберегти

Рис. 22 – Сторінка редагування інформації користувача з рівнем доступу 2 про себе, та співробітників свого підрозділу.

The image shows a web form for user registration. At the top, a red error message reads "Імейл не дійсний" (Email is invalid). The form fields are as follows:

- Департамент : The Faculty of Geography (with a dropdown menu)
- Стан : OPEN (with a dropdown menu showing "OPEN")
- Логін : Gihiruct
- Ім'я : Аркакий (with an input field)
- Прізвище : Леонтев (with an input field)
- По батькові : Васильевич (with an input field)

On the left side, there is a vertical menu with the following buttons:

- Моя сторінка
- Реєстрація нового користувача
- Пошук користувача по базі
- Змінити свій пароль
- Історія користувача
- Вийти

Рис. 23 – Помилка: «Імейл не дійсний.»

Ви не можете змінити рівень адміну на вищий за свій.

Департамент : The Faculty of Geography

Стан : OPEN

Логін : Gihiruct

Ім'я : Аркакий

Прізвище : Леонтев

По батькові : Васильевич

Контрольне питання : -

Моя сторінка

Реєстрація нового користувача

Пошук користувача по базі

Змінити свій пароль

Історія користувача

Вийти

Рис. 24 – Помилка: «Ви не можете змінити рівень адміну на вищий за свій»

Сторінка реєстрації користувача з рівнем прав доступу 2

Користувач з рівнем прав доступу 2 може створити нового користувача, для цього йому необхідно заповнити форму(обов'язковими полями є логін, номер посвідчення, прізвище, ім'я, по батькові та імейл)

(Рис. 25.1-25.2). Пароль користувачу генерується автоматично та надсилається на вказану в базі пошту. Якщо були помилки при введені, форма вкаже, де саме користувач помилився, а результат вводу збережеться. В залежності від вказаної посади буде визначена роль цього користувача, та рівень адміністратора. Імейл перевіряється тестовим повідомленням, за допомогою бібліотеки «email_validate», методу «validate». Також у полі аккаунт «дійсний до» автоматично буде відображатись дата через 4 роки або користувач може заповнити це поле самостійно. Після створення аккаунту логін користувача та час створення запишуться у таблицю історії, а також логін того, хто цього користувача створив. Пароль, буде згенерований випадковим чином та відправлений на вказану, при реєстрації, пошту.

Обробка помилок виконана наступним чином:

- якщо користувач не заповнить будь яке з обов'язкових полів, система повідомить помилку у форматі «Ви не заповнили: перелік полів »(Рис. 26-31)
- користувач може ввести в поле логіну лише англійські символи, числа та знак нижнього підкреслення
- користувач може ввести в поле документу лише цифри та символ слешу
- Імейл не валідний:– Помилка: «Введіть коректний імейл» (Рис.

32)

При успішному створенні користувача, виконується перехід на початкову сторінку (Рис. 21)

The image shows a user registration page. On the left is a sidebar with six blue buttons: "Моя сторінка", "Реєстрація нового користувача", "Пошук користувача по базі", "Змінити свій пароль", "Історія користувача", and "Вийти". The main area has a teal background and contains a registration form with the following fields: "Логін користувача *", "Номер посвідчення *", "Контрольне питання", "Контрольна відповідь", "Прізвище *", "Ім'я *", and "По батькові *". Each field has a corresponding input line. At the bottom right of the form area are three buttons: "Prev" (grey), "Додати" (green), and "Next" (grey).

Рис. 25.1 – Сторінка реєстрації нового користувача

Моя сторінка

Реєстрація нового користувача

Пошук користувача по базі

Змінити свій пароль

Історія користувача

Вийти

Факультет*

Посада

Опис посади

Email

Контактний телефон

Вітальне повідомлення

Аккаунт дійсний до

Prev Додати Next

Рис. 25.2 – Сторінка реєстрації нового користувача

Сторінка реєстрації нового користувача

Логін користувача *

Номер посвідчення *

Контрольне питання

Контрольна відповідь

Прізвище *

Ім'я *

По батькові *

Prev Додати Next

Ви не заповнили Логін, Номер документа, Прізвище, Ім'я, По батькові, Імейл

The image shows a web interface for user registration. On the left, there is a vertical sidebar with seven blue buttons: 'Моя сторінка', 'Реєстрація нового користувача', 'Пошук користувача по базі', 'Змінити свій пароль', 'VPN групи', 'Історія користувача', and 'Вийти'. The main content area has a teal background and a light gray header with the title 'Сторінка реєстрації нового користувача'. Below the header are several input fields with labels: 'Логін користувача *', 'Номер посвідчення *', 'Контрольне питання', 'Контрольна відповідь', 'Прізвище *', 'Ім'я *', and 'По батькові *'. At the bottom of the form are three buttons: 'Prev' (dark gray), 'Додати' (green), and 'Next' (dark gray). A yellow error message box at the bottom center contains the text: 'Ви не заповнили Логін, Номер документа, Прізвище, Ім'я, По батькові, Імейл'.

Рис. 26 – Помилка: Ви не заповнили Логін, Номер посвідчення, Прізвище, Ім'я, По батькові, Імейл

Сторінка реєстрації нового користувача

Логін користувача*
Ferod

Номер посвідчення*

Контрольне питання

Контрольна відповідь

Прізвище*

Ім'я*

По батькові*

Prev Додати Next

Ви не заповнили Номер документа, Прізвище, Ім'я, По батькові, Імейл

Рис. 27 – Помилка: Ви не заповнили Номер посвідчення, Прізвище, Ім'я, По батькові, Імейл

Сторінка реєстрації нового користувача

Логін користувача *
Ferod

Номер посвідчення *
12/30

Контрольне питання

Контрольна відповідь

Прізвище *

Ім'я *

По батькові *

Prev Додати Next

Ви не заповнили Прізвище, Ім'я, По батькові, Імейл

Рис. 28 – Помилка: Ви не заповнили Прізвище, Ім'я, По батькові, Імейл

Сторінка реєстрації нового користувача

Логін користувача*
Ferod

Номер посвідчення*
12/30

Контрольне питання

Контрольна відповідь

Прізвище*
Джавелін

Ім'я*

По батькові*

Prev Додати Next

Ви не заповнили Ім'я, По батькові, Імейл

Рис. 29 – Помилка: Ви не заповнили Ім'я, По батькові, Імейл

Сторінка реєстрації нового користувача

Логін користувача*
Ferod

Номер посвідчення*
12/30

Контрольне питання

Контрольна відповідь

Прізвище*
Джавелін

Ім'я*
Василь

По батькові*

Prev Додати Next

Ви не заповнили По батькові, Імейл

Рис. 30 – Помилка: Ви не заповнили По батькові, Імейл

Сторінка реєстрації нового користувача

Факультет*
new added

Посада
undefined

Опис посади

Email

Контактний телефон

Вітальне повідомлення

Аккаунт дійсний до

Prev Додати Next

Ви не заповнили Імейл

Рис. 31 – Помилка: Ви не заповнили Імейл

Сторінка реєстрації нового користувача

Факультет*
new added

Посада
undefined

Опис посади

Email
qiqumber

Контактний телефон

Вітальне повідомлення

Аккаунт дійсний до

Prev Додати Next

Введіть коректний імейл

The image shows a registration form on a teal background. At the top, a grey header contains the title "Сторінка реєстрації нового користувача". Below the header, there are two dropdown menus: "Факультет*" with the value "new added" and "Посада" with the value "undefined". Below these are five text input fields: "Опис посади", "Email" (containing "qiqumber"), "Контактний телефон", "Вітальне повідомлення", and "Аккаунт дійсний до". At the bottom of the form, there are three buttons: "Prev" (grey), "Додати" (green), and "Next" (grey). Below the buttons is a yellow error message box that says "Введіть коректний імейл".

Рис. 32 – Помилка: Введіть коректний імейл

Код реєстрації нового користувача

```

def register(request):
    if check_cookies(request.COOKIE.get('username'),
request.COOKIE.get('login_status')) \
        and int(request.COOKIE.get('iqlvl')) >= 2:
        error = ""
        if request.method == "POST":
            form = userAllForm(request.POST)
            if (form.data["exp_date"] == ""):
                _mutable = form.data._mutable
                form.data._mutable = True
                form.data["exp_date"] =
datetime.now().date() + timedelta(weeks=288)
                form.data._mutable = _mutable
                _mutable = form.data._mutable
                form.data._mutable = True
                form.data["creator"] =
request.COOKIE.get('username')
                value = request.POST.getlist('title_id')
                lvladmin = UsersAll.adminRole(value[0])
                form.data["admin_level"] = lvladmin
                if (form.data["admin_level"] == 4):
                    form.data["admin"] = "y"
                    form.data["operator"] = "y"
                    form.data["mgroup_id"] = 2
                else:
                    form.data["admin"] = "n"
                    form.data["operator"] = "n"
                    form.data["mgroup_id"] = 1

                form.data["title_id"] = value[0]
                if form.data["title_id"] == "":
                    form.data["title_id"] = 1
                else:
                    form.data["title_id"] = value[0]
                if form.data["dept_id"] == "":
                    form.data["dept_id"] = 1
                form.data["state"] = 'OPEN'
                generated_pass = generatePass()
                password = generated_pass
                password = hashMDPass(password)
                form.data["passwd"] = password
                form.data._mutable = _mutable
                print(form.data)
                if (form.data["login"] != "" and
form.data["login"] != "-"):
                    if (form.data["document_id"] != "" and
form.data["login"] != "-"):
                        if (form.data["lname"] != "" and
form.data["login"] != "-"):

```

```

        if (form.data["fname"] != "" and
form.data["login"] != "-"):
            if (form.data["mname"] != ""
and form.data["login"] != "-"):
                if (form.data["email"] !=
"" and form.data["login"] != "-"):
                    if
(validate(email_address=form.data["email"]) == True):
                        _mutable =
form.data._mutable
                        form.data._mutable
= True
                        for el in
form.data.keys():
                            if
form.data[el] == "":
                                form.data[el] = "-"
                                form.data._mutable
= _mutable
                                if form.is_valid():
                                    users =
UsersAll.objects.filter(login=form.data["login"])
                                    id_dept =
DeptAll.objects.all()
                                    id_title =
TitleAll.objects.all()
                                    dept_number = 1
                                    for el in
id_dept:
                                        if el ==
form.cleaned_data["dept_id"]:
                                            break
                                        dept_number
+= 1
                                            title_number =
1
                                            for el in
id_title:
                                                if el ==
form.cleaned_data["title_id"]:
                                                    break
                                                title_number += 1

form.data._mutable = _mutable

                                if (len(users)
== 0):
                                    formForSave
= UsersAll(login=form.cleaned_data["login"],

```

```
document_id=form.cleaned_data["document_id"],
contr_quest=form.cleaned_data["contr_quest"],
contr_answ=form.cleaned_data["contr_answ"],
lname=form.cleaned_data["lname"].encode(
'cp1251').decode('latin-1'),
fname=form.cleaned_data["fname"].encode(
'cp1251').decode('latin-1'),
mname=form.cleaned_data["mname"].encode(
'cp1251').decode('latin-1'),
dept_id=dept_number,
title_id=title_number,
title=form.cleaned_data["title"].encode(
'cp1251').decode('latin-1'),
email=form.cleaned_data["email"],
phone=form.cleaned_data["phone"],
welcome_msg=form.cleaned_data["welcome_msg"],
exp_date=form.cleaned_data["exp_date"],
admin_level=form.cleaned_data["admin_level"],
admin=form.cleaned_data["admin"],
mgroup_id=form.cleaned_data["mgroup_id"],
operator=form.cleaned_data["operator"],
state=form.cleaned_data["state"],
creator=form.cleaned_data["creator"],
passwd=form.cleaned_data["passwd"],
)
formForSave.save()
```

```

requests.get(
    r =
    f"https://api.unisender.com/ru/api/sendEmail?format=json&api_key
    =6q6o8ud7w8sxs67oga6wedpichx4xogxr8x18uqe&email={form.data['emai
    l']}&sender_name=Support&sender_email=botcreationlab@gmail.com&s
    subject=Your new password.&body=Your new password:
    {generated_pass}&list_id=1")

UsersAll.search_all(
    user =
    UsersAll.objects.filter(login=form.data["login"]))
    history =
    UsersHistory()
    history.user_id = user[0]['id']
    history.date = datetime.now()
    history.action = "CREATION"
    history.creator = request.COOKIES.get('username')
    history.ip
    = get_client_ip(request)
    history.save()
    return
    redirect('home')
    else:
    error =
    "Заповніть ще раз. Такої логін вже існує."
    else:
    error =
    "Перевірте коректність заповнення форми."
    else:
    error = "Введіть
    коректний імейл"

    req_params = ['login', 'document_id', 'lname',
    'fname', 'mname', 'email']
    tables = []
    for i in req_params:
        print(i)
        print(form.data[i])
        if form.data[i] == '':
            tables.append(i)

    if error == "":
        error = "Ви не заповнили " +
    '/' .join(tables)
    _mutable = form.data._mutable

```

```
        form.data._mutable = True
        form.data["exp_date"] = ""
        form.data._mutable = _mutable
        data = {
            'form': form,
            'lvl': int(request.COOKIES.get('iqlvl')),
            'error': error
        }
        return create_cookies(request,
request.COOKIES.get('iqlvl'), request.COOKIES.get('username'),
                                'main/registration.html',
data)
```

```
        form = userAllForm()
        data = {
            'form': form,
            'lvl': int(request.COOKIES.get('iqlvl')),
            'error': error
        }
        return create_cookies(request,
request.COOKIES.get('iqlvl'), request.COOKIES.get('username'),
                                'main/registration.html',
data)
```

```
    else:
        return delete_cookies()
```

4.4. Сторінка пошуку

На сторінці пошуку Рис. 33 користувач може здійснити пошук по одному або більше критеріям. Якщо результат буде більш ніж 5 користувачів, система попросить уточнити пошук. Після цього знизу під формою з'явиться результат пошуку. Рис. 34

Також реалізовано розподіл прав, якщо у користувача рівень адміністрування 2, то він зможе знаходити та редагувати лише тих співробітників, що знаходяться у нього в підрозділі.

Користувач з рівнем адміністрування 3 має усі права користувача з рівнем 2 та додатково зможе знаходити та редагувати всіх користувачів, що знаходяться в системі.

На цій сторінці опрацьовані наступні помилки:

- Якщо нічого не введено:– Помилка: Уведіть хоча б один із пунктів для пошуку (Рис. 35)
- Якщо більше 5 результатів:– Помилка: Уточніть запит (Рис. 36)
- Помилка: За такими даними користувача не знайдено. (Рис. 37)

The image shows a web interface for searching users. On the left is a sidebar with seven blue buttons: 'Моя сторінка', 'Реєстрація нового користувача', 'Пошук користувача по базі', 'Змінити свій пароль', 'VPN групи', 'Історія користувача', and 'Вийти'. The main area has a teal background. At the top, a grey header contains the title 'Сторінка пошуку користувачів'. Below it, a dark grey box contains the instruction 'Будь ласка, заповніть хоча б одне з полів'. The search form consists of six input fields: 'Логін користувача', 'Номер посвідчення', 'Прізвище', 'Ім'я', 'По батькові', and 'Email'. A green 'Пошук' button is positioned at the bottom right of the form.

Сторінка пошуку користувачів

Будь ласка, заповніть хоча б одне з полів

Логін користувача

Номер посвідчення

Прізвище

Ім'я

По батькові

Email

Пошук

Рис. 33 – Сторінка пошуку інформації о користувачі

Логін	Прізвище	Ім'я	Побатькові	Факультет	Стан	Дійсний до
Gihiruct	Леонтев	Аркадій	Васильевич	The Faculty of Geography	OPEN	11-12-2027
			Редагувати			

Рис. 34 – Вивід інформації про користувача.

Сторінка пошуку користувачів

Будь ласка, заповніть хоча б одне з полів

Логін користувача

Номер посвідчення

Прізвище

Ім'я

По батькові

Email

Пошук

Уведіть хоча б один із пунктів для пошуку

The image shows a user search interface on a teal background. At the top, a grey header contains the title 'Сторінка пошуку користувачів'. Below it, a dark grey box with white text says 'Будь ласка, заповніть хоча б одне з полів'. The form consists of six input fields: 'Логін користувача', 'Номер посвідчення', 'Прізвище', 'Ім'я', 'По батькові', and 'Email'. Each field is followed by a horizontal line. Below the fields is a green 'Пошук' button. At the bottom, a yellow error message box states 'Уведіть хоча б один із пунктів для пошуку'.

Рис. 35 – Помилка: Уведіть хоча б один із пунктів для пошуку

Сторінка пошуку користувачів

Будь ласка, заповніть хоча б одне з полів

Логін користувача

Номер посвідчення

Прізвище

Ім'я

По батькові

Email

jovoxi3679@musezoo.com

Пошук

Уточніть запит

Рис. 36 – Помилка: Уточніть запит

Сторінка пошуку користувачів

Будь ласка, заповніть хоча б одне з полів

Логін користувача
dnjankj

Номер посвідчення

Прізвище

Ім'я

По батькові

Email

Пошук

За такими даними користувача не знайдено.

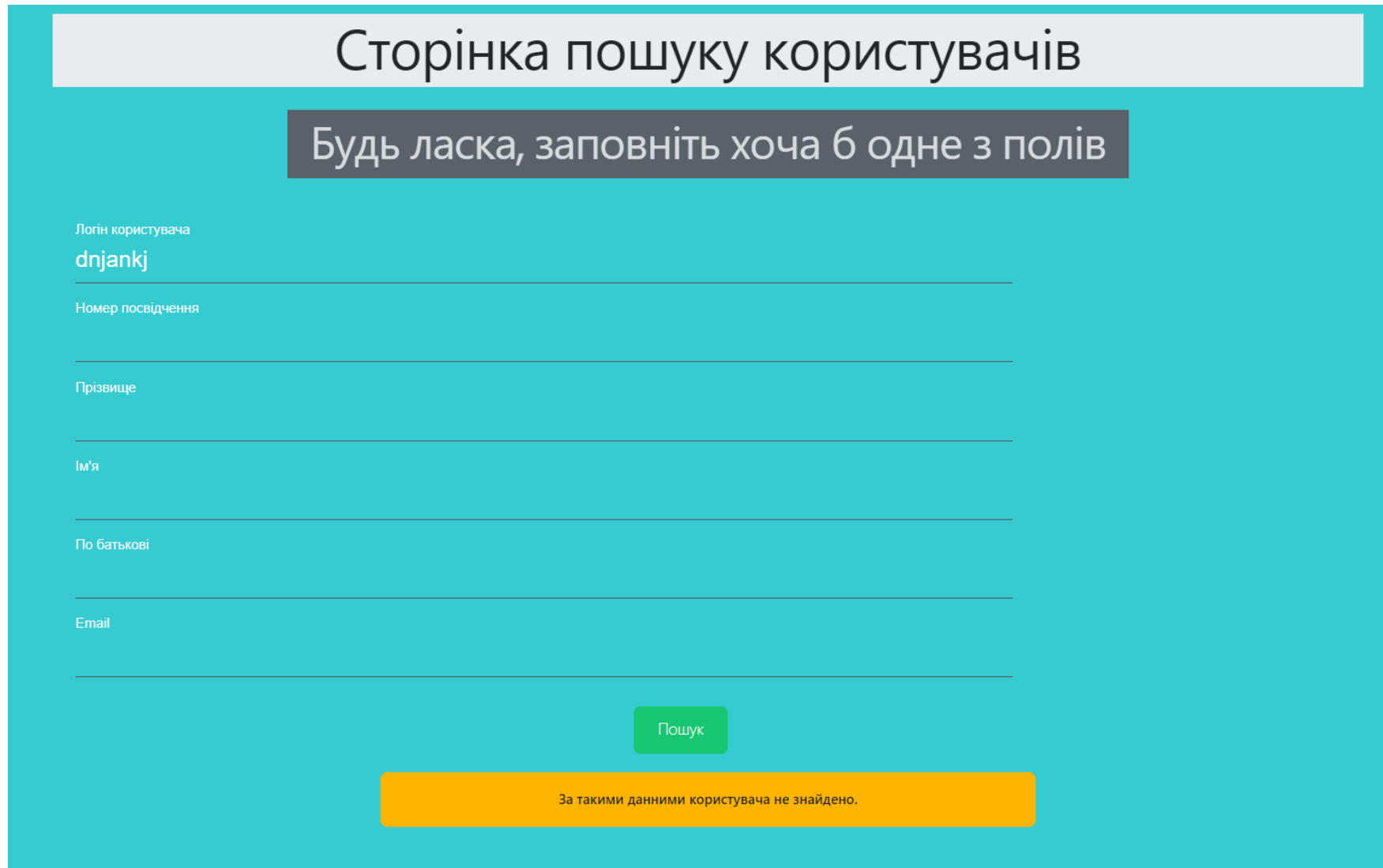
The image shows a user search interface on a teal background. At the top, a grey header contains the title 'Сторінка пошуку користувачів'. Below it, a dark grey box contains the instruction 'Будь ласка, заповніть хоча б одне з полів'. The search form consists of six input fields: 'Логін користувача' (with the value 'dnjankj'), 'Номер посвідчення', 'Прізвище', 'Ім'я', 'По батькові', and 'Email'. A green 'Пошук' button is positioned below the fields. At the bottom, a yellow error message states 'За такими даними користувача не знайдено.'

Рис. 37 – Помилка: За такими даними користувача не знайдено.

Код виводу інформації про користувача

```

def find(request):
    if check_cookies(request.COOKIES.get('username'),
request.COOKIES.get('login_status')) \
        and int(request.COOKIES.get('iqlvl')) >= 2:
        users = ""
        error = ""
        info_show = False
        searching_users = []
        if request.method == "POST":
            form = userAllForm(request.POST)
            if (form.data["fname"] != "" or
form.data["login"] != "" or form.data["document_id"] != ""
                or form.data["lname"] != "" or
form.data["mname"] != "" or form.data["email"] != ""):
                if int(request.COOKIES.get('iqlvl')) == 2:
                    users = UsersAll.search_all(
UsersAll.objects.filter(fname__icontains=form.data["fname"].enco
de('cp1251').decode('latin-1'))

.filter(login__icontains=form.data["login"])

.filter(document_id__icontains=form.data["document_id"])

.filter(lname__icontains=form.data["lname"].encode('cp1251').dec
ode('latin-1'))

.filter(mname__icontains=form.data["mname"].encode('cp1251').dec
ode('latin-1'))

.filter(email__icontains=form.data["email"])
                    .filter(dept_id =
UsersAll.objects.get(login =
request.COOKIES.get('username')).dept_id)
                    else:
                        users = UsersAll.search_all(
UsersAll.objects.filter(fname__icontains=form.data["fname"].enco
de('cp1251').decode('latin-1'))

.filter(login__icontains=form.data["login"])

.filter(document_id__icontains=form.data["document_id"])

.filter(lname__icontains=form.data["lname"].encode('cp1251').dec
ode('latin-1'))

.filter(mname__icontains=form.data["mname"].encode('cp1251').dec
ode('latin-1'))

```


4.5. Зміна паролю

На сторінці зміни паролю (Рис. 38), користувач може змінити свій пароль. Заповнив поля пароля та підтвердження паролю. Пароль перевіряється на кількість символів (мінімум 8), та порівнюється з підтвердженням. Якщо все добре – записується у базу.

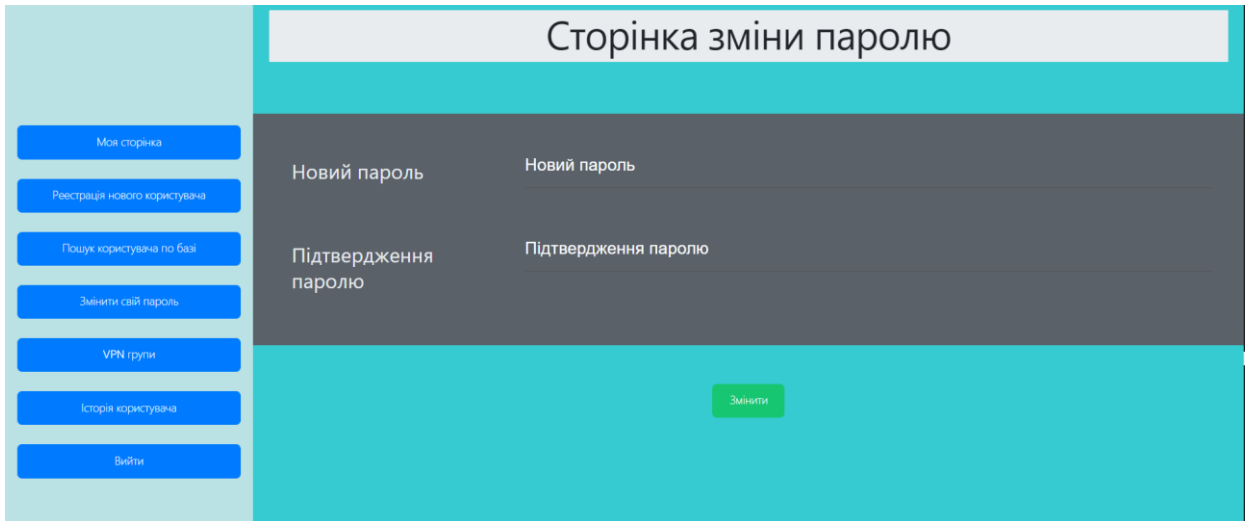


Рис. 38 – Сторінка зміни паролю користувача

На цій сторінці опрацьовані наступні помилки:

- Поле не заповнене:– Помилка: Заповніть це поле (Рис. 39)
- Якщо пароль менше 8 символів:– Помилка: Мінімальна довжина паролю: 8 символів. (Рис. 40)
- Паролі не співпадають: – Помилка: Паролі не співпадають. (Рис. 41)

Сторінка зміни паролю

Новий пароль

Підтвердження паролю

Новий пароль

Підтвердження паролю

Заполните это поле

Рис. 39 – Помилка: Заповніть це поле

Сторінка зміни паролю

Новий пароль

Підтвердження паролю

Новий пароль

Підтвердження паролю

Мінімальна довжина паролю 8 символів.

Змінити

Рис. 40 – Помилка: Мінімальна довжина паролю: 8 символів.

Сторінка зміни паролю

Новий пароль

Підтвердження паролю

Новий пароль

Підтвердження паролю

Паролі не співпадають.

Змінити

Рис. 41 – Помилка: Паролі не співпадають.

Код зміни паролю користувача

```

        if check_cookies(request.COOKIE
S.get('username'),
request.COOKIE
S.get('login_status'))\
            and int(request.COOKIE
S.get('iqlvl')) >= 2:
            success = ""
            history = UsersHistory.objects.all()
            if request.method == "POST":
                form = userAllForm(request.POST)
                users =
UsersAll.search_all(UsersAll.objects.filter(login=request.COOKIE
S.get('username')))
                if form.data["password"] ==
form.data["repassword"]:
                    if (len(form.data["password"]) >= 8):
                        password = form.data["password"]
                        password = hashMDPass(password)
                        user = UsersAll(user_id=users[0]["id"])
                        user.passwd = password
                        success = "Пароль успішно змінено."
                        history = UsersHistory()
                        history.user_id = users[0]["id"]
                        history.date = datetime.now()
                        history.action = "CHPASS"
                        history.creator =
request.COOKIE
S.get('username')
                        history.ip = get_client_ip(request)
                        history.save()
                        user.save(update_fields=['passwd'])
                    else:
                        success = "У паролі повинно бути хоча б
8 символів."
                else:
                    success = "Паролі не співпадають."
            form = change_passForm()
            data = {
                'form': form,
                'success': success,
                'lvl': int(request.COOKIE
S.get('iqlvl'))
            }
            return create_cookies(request,
request.COOKIE
S.get('iqlvl'), request.COOKIE
S.get('username'),
                                'main/ownchange
pass.html',
data)

    else:
        return delete_cookies()

```

4.6. Історія акаунту

На сторінці перевірки історії акаунта користувача перевіряються куки файли, якщо їх немає, то перенаправляє на вхід. Якщо є куки файли, то користувач може ввести логін користувача та натиснути кнопку «пошук», після натискання кнопки, на сторінці відображається таблиця з даними про події за цим логіном. У форматі: дата події, подія, причина, оператор, IP-адреса та кнопка детальніше. (Рис. 42.1-42.2). Користувач з рівнем адміністратора 2 зможе бачити лише історію користувачів свого підрозділу.

The image shows a web interface for searching user account history. On the left is a vertical sidebar with seven blue buttons: 'Моя сторінка', 'Реєстрація нового користувача', 'Пошук користувача по базі', 'Змінити свій пароль', 'VPN групи', 'Історія користувача', and 'Вийти'. The main content area has a teal background. At the top, a grey header contains the title 'Пошук історії аккаунту користувача'. Below it, a teal banner reads 'Для отримання історії користувача введіть логін'. A dark grey input field is labeled 'Логін користувача' and contains the text 'Логін'. A green 'Пошук' button is positioned below the input field.

Рис. 42.1 – Сторінка для пошуку історії аккаунту користувача

<p>Моя сторінка</p> <p>Реєстрація нового користувача</p> <p>Пошук користувача по базі</p> <p>Змінити свій пароль</p> <p>VPN групи</p> <p>Історія користувача</p> <p>Вийти</p>	21:05 06-06-2022	CHPASS	None	nldfnkj	95.158.53.167	Детальніше
	11:18 07-06-2022	CHDATA	Changed VPN group	nldfnkj	95.158.53.167	Детальніше
	11:18 07-06-2022	CHDATA	['Control question changed from лотт лоло to 11...	nldfnkj	95.158.53.167	Детальніше
	11:19 07-06-2022	CHDATA	['Exp. date changed from 2022-12-31 00:00:00	nldfnkj	95.158.53.167	Детальніше

Рис. 42.2 – Результат пошуку історії акаунту користувача

У полі причина, записуються зміни історії акаунту користувача. (Рис. 43) Якщо змін було багато, текст показує 47 символів і три крапки, та можна перейти за посиланням для отримання повної інформації про обраний запис.

Дата	Подія	Причина	Оператор	ip	Перегляд
23:22 02-06-2022	CREATION	None	bloodelf	127.0.0.1	Детальніше
01:25 03-06-2022	CHPASS	None	bloodelf	127.0.0.1	Детальніше
14:31 04-06-2022	CHDATA	['Email changed from talabor548@nzaif.com to jo...']	kmcdma	127.0.0.1	Детальніше

Дата
04-06-2022
Дія
CHDATA
Оператор
kmcdma
Зміни
'Email changed from talabor548@nzaif.com to jovoxi3679@musezoo.com'
Айпі адреса
127.0.0.1

Рис. 43 – Зміни історії акаунту користувача

Код пошуку історії акаунту користувача

```

def history(request):
    if check_cookies(request.COOKIES.get('username'),
request.COOKIES.get('login_status')) \
        and int(request.COOKIES.get('iqlvl')) >= 2:
        info = []
        error = ""
        info_show = False
        if request.method == "POST":
            form = userAllForm(request.POST)
            login = form.data["login"]
            if (login != ""):
                logD = UsersAll.objects.all()
                idOfuser = -1
                for el in logD:
                    if (login == el.login):
                        idOfuser = el.user_id
                        break
                if (idOfuser != -1):
                    history = UsersHistory.objects.all()
                    for el in history:
                        if (idOfuser == el.user_id):
                            if el.reason != None:
                                if len(el.reason) > 50:
                                    el.reason = "Дивись
детальніше"
                                info.append({
                                    "date":
el.date.strftime("%H:%M %d-%m-%Y"),
                                    "action": el.action,
                                    "creator": el.creator,
                                    "reason": el.reason,
                                    "ip": el.ip,
                                    'id': el.id
                                })
                            if info == []:
                                error = "Історії за цим
користувачем не знайдено"
                            else:
                                info_show = True
                        else:
                            error = "За таким логіном користувача
не знайдено"
                    else:
                        error = "Ви не ввели логін"

        form = userAllForm()
        data = {
            'form': form,
            'info': info,
            'error': error,

```

```
        'lvl': int(request.COOKIES.get('iqlvl')),
        'show': info_show
    }
    return create_cookies(request,
request.COOKIES.get('iqlvl'), request.COOKIES.get('username'),
        'main/history.html', data)
else:
    return delete_cookies()
```

При натисканні кнопки вийти (Рис. 44) – куки видаляються, а користувача переадресовує на сторінку входу. Рис. 3

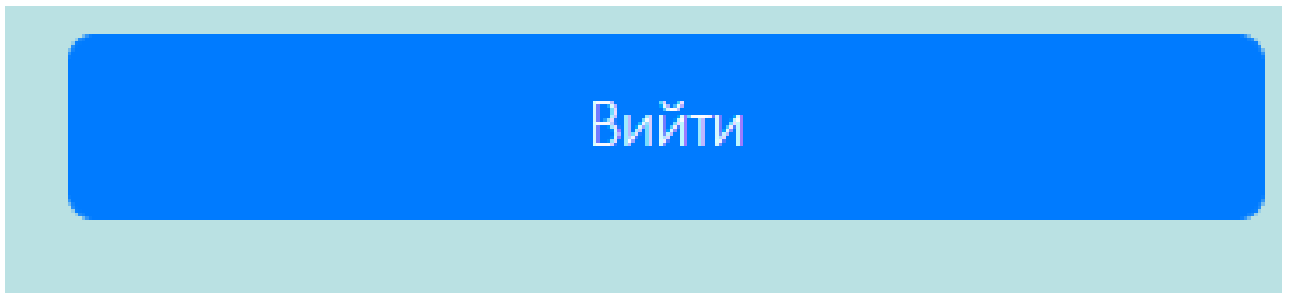


Рис. 44 -- Вигляд кнопки вийти.

Код

```
def logout(request):
    return delete_cookies()
```

4.7. Додавання користувачів до VPN груп.

Для користувачів у котрих рівень адміністратора більший або рівний за 4, доступна функція адміністрування VPN доступу:

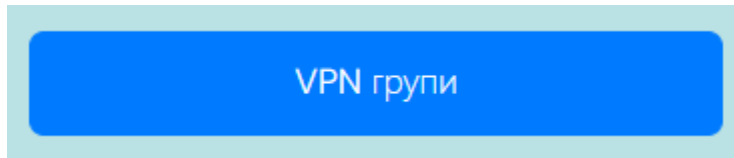


Рис. 45 – Кнопка переходу до налаштування VPN доступу.

Коли користувач заходить у це меню, перевіряється його рівень адміністратора, при рині 4, йому показується на сторінці лише група VPN та у нього є можливість додати користувача до неї: У полі показується лише останні 5 доданих записів. (Рис. 46)

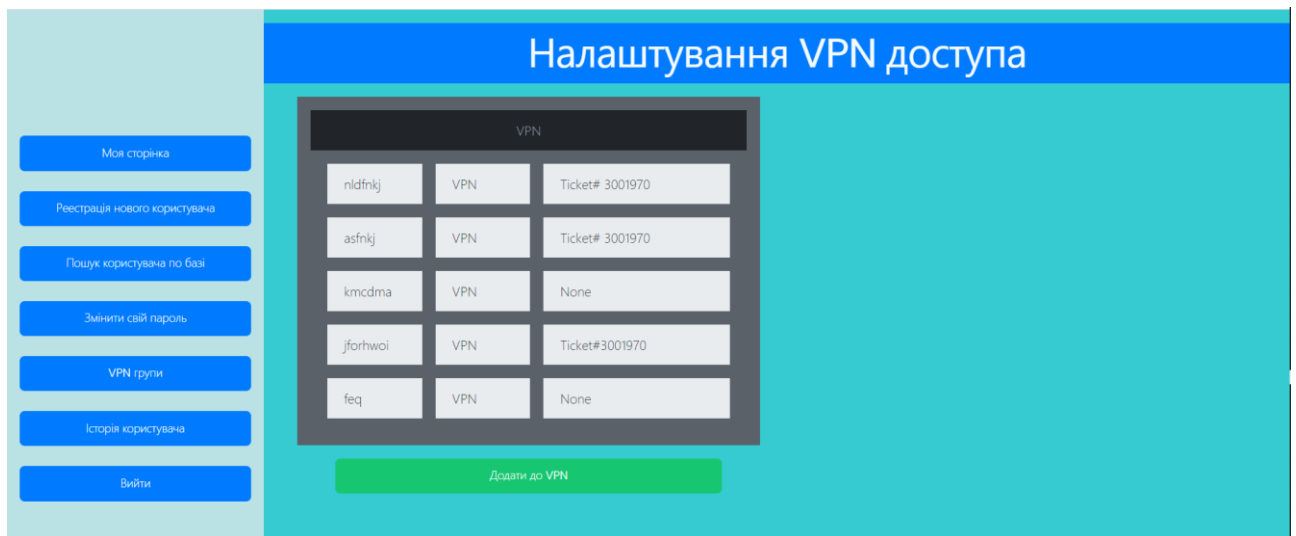


Рис. 45 – Налаштування VPN доступу

При додаванні до групи VPN користувач перенаправляється на сторінку форми пошуку користувачів. (Рис. 46) На цій сторінці, користувач може знайти людину та, якщо її ще не має у цій VPN групі, додати її. Результатом пошуку може бути лише одна людина, тобто до того моменту, коли не буде введено достатньо інформації, додати нового користувача VPN групі буде неможливо.

Будь ласка, заповніть хоча б одне з полів

Логін користувача

Номер посвідчення

Прізвище

Ім'я

По батькові

Email

Рис. 46 – Форма пошуку користувачів для додавання у групу VPN доступу

Знайденого користувача можливо додати до групи VPN доступу шляхом натискання кнопки «Додати». (Рис. 47)

Так при пошуку ми можемо додати:

Додати користувача	Логін	Прізвище	Ім'я	Побатькові	Факультет
<input type="button" value="Додати"/>	bloodelf	Лукашов	Олександр	Валентинович	The Faculty of Radiophysics

Рис. 47 -- Додавання до групи VPN доступу користувача.

Якщо ж користувач вже є у групі, то буде виведено (Рис. 48) так:

Логін	Прізвище	Ім'я	Побатькові	Факультет
jforhwoi	kjvfwo	kdnov	kvndkl	The Faculty of Chemistry

Рис. 48 – Користувач вже є у групі доступу.

Користувач рівень доступу 5

Якщо у користувача рівень доступу 5, то він крім VPN доступу додатково бачить поле VPN-NOC доступу та може додавати туди користувачів. Він бачить лише останні 5 записів. (Рис. 49).

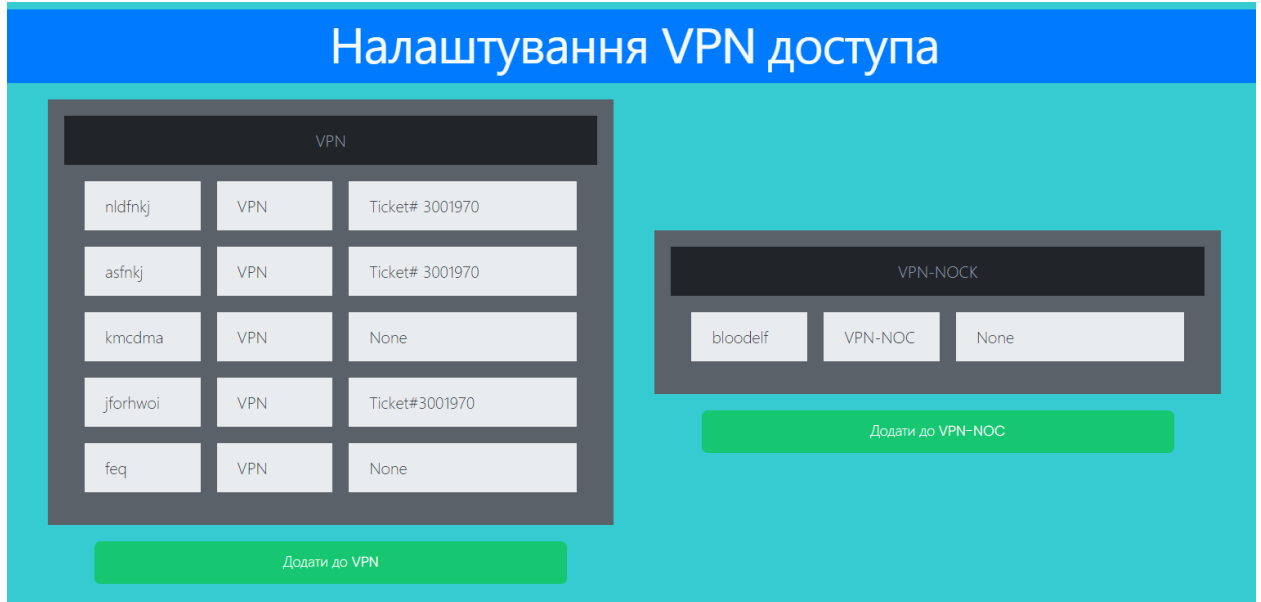


Рис. 49 – Налаштування VPN та VPN-NOC доступу

Додавання користувача VPN-NOC доступу, реалізовано так само як і додавання користувача VPN доступу.

4.8. Відновлення паролю.

Якщо користувач при вході в систему забув пароль, він може створити новий.

Для цього йому необхідно перейти з початкової сторінки входу до web панелі (Рис. 3) до форми відновлення паролю (Рис. 50).

Рис. 50 – Форма відновлення паролю

Для відновлення необхідно ввести логін користувача та набрати код на панелі з правої сторони. Після вводу логіна та коду користувачу за вказаним логіном, у базі шукається поштовий запис та на нього відправляється посилання для зміни паролю. Воно дійсне протягом 15 хвилин.

Обробтані помилки:

- Не введено логін – Помилка: «Заповніть це поле.» (Рис. 51)
- Введено не існуючий логін– Помилка: «Такого логіну не існує.» (Рис. 52)
- Логін існує, але не правильно набраний код– Помилка: «Неправильно набраний код.» (Рис. 53)

Після правильного введення логіна та унікального коду, нас перенаправляє на сторінку входу і показує, що пароль повинен прийти на пошту.

Для відновлення паролю ведіть логін та код перевірки

Логін

Заполните это поле.
Будь ласка, введіть код для підтвердження, що ви не робот

77921

Змінити пароль

2	9	delete
7	4	5
1	6	/
0	8	3

Рис. 51 – Помилка: Заповніть це поле.

Для відновлення паролю ведіть логін та код перевірки

Логін

jdknka

Будь ласка, введіть код для підтвердження, що ви не робот

20248

Змінити пароль

Такого логіну не існує.

7	9	delete
0	2	3
5	6	8
4	1	/

Рис. 52 – Помилка: Такого логіну не існує.

Для відновлення паролю ведіть логін та код перевірки

Логін

bloodelf

Будь ласка, введіть код, для підтвердження, що ви не робот

19686

Змінити пароль

Не правильно набраний код.

delete	2	7
1	/	9
0	3	4
5	6	8

Рис. 53 – Помилка: Неправильно набраний код.

Після правильного введення логіна та капчі, нас перенаправляє на сторінку входу і показує, що пароль повинен прийти на пошту (Рис. 54).

Логін

Пароль

Увійти

Згадати пароль

Посилання на відновлення паролю, відправлено на пошту

Рис. 54 – Посилання на відновлення паролю, відправлено на пошту.

Після цього на пошту надходить посилання (Рис. 55)

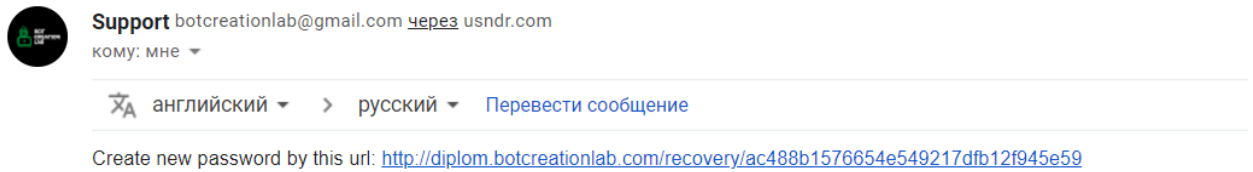


Рис. 55 — Посилання для переходу на сайт для відновлення паролю.

Перехід за посиланням направляє до форми зміни паролю (Рис. 56) де він може задати власний пароль:

Рис. 56 – Форма зміни паролю,

На цій формі оброблені такі ж самі помилки, що і в формі зміни паролю у користувача. А також додано контроль за часом, після 15 хвилин буде напис, що ця форма вже не дійсна.

Рис. 57 – Час зміни паролю сплив.

ВИСНОВКИ

У ході виконання дипломного проекту було:

- Проаналізовано програмне забезпечення інтерфейсу адміністрування співробітників організації КНУ Шевченка, та виявлена необхідність модернізації програмного забезпечення та розширення його функціоналу.
- Описано алгоритм роботи наявного програмного забезпечення інтерфейсу адміністрування співробітників.
- Для розробки нового інтерфейсу програмного забезпечення роботи з базою адміністрування співробітників університету узгоджено зміни вимог до програмного забезпечення .
- Розроблено згідно узгоджених вимог алгоритм роботи інтерфейсу програмного забезпечення адміністрування співробітників університету.
- Виконано розробку програмного забезпечення.

Завдання дипломного проекту на розробку інтерфейсу адміністрування співробітників організації КНУ Шевченка виконане в повному обсязі згідно узгоджених вимог.

Вважаю, що розроблене програмне забезпечення є дуже актуальним сьогодні та його можливо використати як частину нового програмного забезпечення, в рамках необхідної та своєчасної модернізації цифрової інфраструктури КНУ Шевченка

Аналіз наявного стану програмного забезпечення роботи, системи адміністрування співробітників організації КНУ Шевченка, з наданою базою даних свідчить о необхідності проведення модернізації системи управління базою даних, та зміни в структурі самої бази даних.

Сучасні вимоги до експлуатації баз даних в рамках однієї організації свідчать що економічно вигідніше експлуатувати, обслуговувати, модернізувати та адмініструвати єдину базу даних замість декількох.

Пропоную дати можливість студентам взяти участь у актуальній сьогодні програмі цифрової модернізації інфраструктури КНУ Шевченка шляхом впровадження до учбової програми лекцій, та завдань на тему створення сучасної системи бази даних та її адміністрування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Введите пароль: обзор форм авторизации и альтернативных способов идентификации пользователей [Электронный ресурс] // Офтоп. – 2016. – Режим доступа до ресурсу: <https://vc.ru/flood/17468-authentication-state>.
2. ПОПУЛЯРНЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ LINUX [Электронный ресурс] // losst. – 2020. – Режим доступа до ресурсу: <https://losst.ru/populyarnye-yazyki-programmirovaniya-linux/>.
3. Исак. Лучшие языки программирования для разработчиков Linux [Электронный ресурс] / Исак – Режим доступа до ресурсу: <https://www.linuxadictos.com/ru/mejores-lenguajes-programacion-desarrolladores-linux.html>.
4. Программирование под Linux [Электронный ресурс] // pro-spo – Режим доступа до ресурсу: <https://pro-spo.ru/linuxprog>.
5. Django documentation [Электронный ресурс] // The Django Software Foundation. – 2022. – Режим доступа до ресурсу: <https://docs.djangoproject.com/en/4.0/>.
6. Форум програмістів [Электронный ресурс] – Режим доступа до ресурсу: <https://stackoverflow.com>.
7. Внешние ключи FOREIGN KEY [Электронный ресурс] // Metanit. – 2019. – Режим доступа до ресурсу: <https://metanit.com/sql/mysql/2.5.php>.
8. Syncdb vs Migration [Электронный ресурс] // Ruddra.com. – 2014. – Режим доступа до ресурсу: <https://ruddra.com/syncdb-vs-migration/>.
9. Python – Django Models (1054, “Unknown column in ‘field list’”) [Электронный ресурс] // iTecNote – Режим доступа до ресурсу: <https://itecnote.com/tecnote/python-django-models-1054-unknown-column-in-field-list/>.
10. В input разрешить ввод только чисел через jQuery [Электронный ресурс] // Blogjquery. – 2018. – Режим доступа до ресурсу: <https://blogjquery.ru/v-input-razreshit-vvod-tolko-chisel-cherez-jquery/>.

11. How to Get Current Date & Time in JavaScript [Электронный ресурс] // TecAdmin. – 2018. – Режим доступа до ресурсу: <https://tecadmin.net/get-current-date-time-javascript/>.
12. Универсальный декодер - конвертер кириллицы [Электронный ресурс] – Режим доступа до ресурсу: <https://2cyr.com/decode/?lang=ru>.
13. If statement in Django template [Электронный ресурс] // PythonGuides. – 2021. – Режим доступа до ресурсу: <https://pythonguides.com/if-statement-in-django-template/>.
14. КАК ПРЕОБРАЗОВАТЬ СТРОКУ ИЗ CP-1251 В UTF-8? [Электронный ресурс] // Ru Python – Режим доступа до ресурсу: <https://www.rupython.com/cp-1251-utf-8-19779.html>.
15. Руководство часть 9: Работа с формами [Электронный ресурс] // mdn web docs. – 2021. – Режим доступа до ресурсу: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Forms>.
16. Модуль datetime [Электронный ресурс] // all-python – Режим доступа до ресурсу: <https://all-python.ru/osnovy/modul-datetime.html>.
17. Shards UI [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: <https://designrevision.com/demo/shards/>.
18. Уроки Django [Электронный ресурс] // itProger. – 2020. – Режим доступа до ресурсу: <https://itproger.com/course/django>.
19. Get started with Bootstrap [Электронный ресурс] – Режим доступа до ресурсу: <https://getbootstrap.com/docs/4.0/getting-started/introduction/>.
20. Станьте гуру в Git. [Электронный ресурс] // atlassian – Режим доступа до ресурсу: <https://www.atlassian.com/ru/git/tutorials>.
21. Глава 1. Введение в Django [Электронный ресурс] – Режим доступа до ресурсу: <https://djbook.ru/ch01.html>.
22. Источник: <https://pythonstart.ru/string/lower-upper-python> [Электронный ресурс] // pythonstart. – 2020. – Режим доступа до ресурсу: <https://pythonstart.ru/string/lower-upper-python>.

23. MD5 hash in Python [Электронный ресурс] // geeksforgeeks. – 2020.
– Режим доступа до ресурсу: <https://www.geeksforgeeks.org/md5-hash-python/>.