

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

ВИПУСКНА РОБОТА

**на тему:
«Системи автоматизації розселення
студентів по гуртожиткам»**

Завідувач

випускаючої кафедри

Довбиш А.С.

Керівник роботи

Власенко О. В.

Студента групи ІН – 71

Скороход А. А.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2021 р.

ЗАВДАННЯ

до випускної роботи

Студента четвертого курсу, групи ІН-71, спеціальності «122 – Комп'ютерні науки» денної форми навчання Скорохода Андрія Анатолійовича.

Тема: «Система автоматизації розселення студентів по гуртожиткам»

Затверджена наказом по СумДУ

№ _____ від _____ 2021 р.

Зміст пояснювальної записки: 1) інформаційний огляд; 2) вибір методів рішення задачі; 3) практична реалізація; 4) висновки до роботи.

Дата видачі завдання “ _____ ” _____ 2021 р.

Керівник випускної роботи _____ Власенко О. В.

Завдання прийняв до виконання _____ Скороход А. А.

РЕФЕРАТ

Записка: 77 сторінок, 30 рисунків, 9 джерел

Об'єкт дослідження – автоматизація прикладних бізнес-задач на прикладі розселення студентів по гуртожиткам

Мета роботи – розробка системи для автоматизації розселення студентів по гуртожиткам.

Методи дослідження – багатосторінковий сайт на основі технологій PHP, JavaScript, MySQL, jQuery, HTML5/CSS3.

Результати – створено багатосторінковий веб-сайт для автоматизації розселення студентів по гуртожиткам, де було створено можливість розселювати студентів, перевіряти факт їх проживання в гуртожитку, формувати звітність в вигляді Excel та особистий кабінет для мешканців гуртожитку. Сайт готовий для деплою на серверах Сумського державного університету.

ВЕБ-ДОДАТОК, АВТОМАТИЗАЦІЯ, WEB, FULL-STACK,
PHP, JAVASCRIPT, JQUERY

ЗМІСТ

ВСТУП	5
1. ІНФОРМАЦІЙНИЙ ОГЛЯД	6
1.1. Огляд існуючих рішень	6
1.2. Постановка задачі.....	7
2. ВИБІР МЕТОДУ РІШЕННЯ	7
2.1 Вибір технологій.....	9
2.2 Опис технологій для клієнтської частини	9
2.3 Опис технологій для серверної частини	10
2.4 Інші використані технології	11
3. ПРОГРАМНА РЕАЛІЗАЦІЯ	13
3.1 Створення бази даних	13
3.2 Реалізація клієнтської та серверної частин.....	14
3.3 Інструкція користувача для адміністратора	16
3.4 Інструкція користувача для вахтера.....	22
3.5 Інструкція користувача для студента	28
ВИСНОВОК.....	31
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	32
ДОДАТОК А – ЛІСТИНГ ПРОГРАМИ	33

ВСТУП

Автоматизація все більше і більше проникає в численні галузі людської діяльності. Причин для введення автоматизації безліч, але, насамперед, це уникнення важкої або монотонної роботи та збільшення прибутку. Звільнення людей від подібної роботи не тільки дозволяє їм переключитися на задачі, що вимагають більш нестандартного підходу, а й дозволяє зекономити підприємству велику кількість людино-годин [1]. Автоматизація може бути де завгодно: від обліку відвідування занять до тестування програмного забезпечення; від банального скрипта, що видаляє дублікати в Excel-файлі, до складних RPA-систем, що імітують роботу людей, використовуючи навіть технології штучного інтелекту. RPA розшифровується як “Robot Process Automation”, але не має нічого спільного з механічними роботами. Це просто програми (своєрідні боти), які, де-факто, виконують людську роботу, але швидше, з нульовою кількістю виключно людських помилок (внаслідок таких факторів як неуважність, сонність і т.д.), та безперервно [1].

Звісно, на поприщі автоматизації можна розводити диспути про позбавлення людей робочих місць, але це не стосується автоматизації в тому вигляді, в якому вона задіяна під час проектування та створення системи автоматизації розселення студентів, яка і є темою даної роботи. У контексті даної системи це не повна заміна людини, що займається розселенням студентів, а лише інструмент, що допоможе уникнути типових задач, що неодноразово повторюються.

Актуальність роботи насамперед полягає, у вирішенні конкретної практичної задачі. Процес автоматизації розселення студентів та подальший контроль проживання в гуртожитку, є особливо актуальним під час поломок пропускних турнікетів. Додана можливість студентам відслідковувати рейтингові бали, які є необхідними для поселення.

1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1. Огляд існуючих рішень

На даний момент процес поселення відбувається наступним чином. Інформація опрацьовується та доноситься до студентів у вигляді так званих “шахматок” – Excel-файлів, що містять інформацію про кімнати та їх мешканців (див. рис. 1.1).

	A	B	C	D	E	F	G
1					Гуртожиток * Секція ** факультет ****		
2	№ п/п	№ ордеру	Рейтинг	Кімната	ПІБ	Курс	Телефон
3	1			69			
4	2						
5	3		50		Владова *****	*	
6	4		50		Іванченко *****	*	
7	5		50	69	Краповий *****	*	
8	6		50		Кубінець *****	*	
9	7					*	

Рисунок 1.1 – Зразок “шахматки” (всі особисті дані замінені)

Було вирішено розробити систему автоматизації, через те що:

- розроблена система дозволяє наочніше візуалізувати процес розселення;
- “шахматки” зручні для донесення інформації, але не надто зручні для розселення студентів по кімнатах, тому сам процес розселення змінений, але за необхідності система має експорт до Excel-файлу;
- реляційна база даних дозволяє краще обробляти дані та надійніше їх зберігати за умов попереднього страхування (наприклад, дамів бази);
- додано профіль мешканця, де він має змогу переглянути інформацію про себе, деталі проживання в гуртожитку та свій рейтинговий бал;
- додано профіль вахтера для перевірки факту проживання студента в гуртожитку.

Чи можна було якось обійтися без створення цілої системи? Можна було б автоматизувати все, наприклад, застосувавши вище згадану RPA, наприклад, скориставшись ботами від Automation Anywhere. Automation Anywhere – це платформа для роботизації та автоматизації бізнес-процесів, яка є провідною та визнаною одним зі світових лідерів на ринку RPA-рішень. Automation Anywhere дозволяє організаціям автоматизувати процеси, якими зазвичай займаються працівники. Це веб-орієнтована система [5], яка використовує Контрольну кімнату (сервер з ботами) для запуску автоматизованих завдань. Дана платформа дозволяє створити спеціального бота, який по суті є звичайною програмою. Такий бот, створений на платформі Automation Anywhere Enterprise (AAE) може запам'ятати дії користувача, дозволяє вносити корективи та потім повторювати їх. Існує три режими запису команд: запис лише того, що відбувається в браузері (Web recorder), запис всього, що відбувається на екрані (Screen Recorder) та розумний запис, який комбінує два попередні способи (Smart recorder). Концепт роботи подібний до розширення Selenium для браузера. Але мінуси даного підходу в тому, що рішення про розселення приймає людина, тобто, потрібні логічні висновки про правильність вибору мешканців для кімнати. Також подібне програмне забезпечення є платним.

1.2. Постановка задачі

Для зручності, вимоги до системи автоматизації згрупуємо за ролями користувачів:

Студент (Student):

- перегляд персональної інформації;
- перегляд подробиць про місце проживання;
- відслідковування рейтингового балу, що відображається червоним, якщо він є нижчим за норму.

Вахтер (Watchman):

- перевірка факту проживання конкретного студента в гуртожитку;

- пошук персональної інформації про мешканця.

Адміністратор (Admin):

- додавання, видалення та редагування гуртожитків/секцій/кімнат у створеній базі даних;
- додавання, видалення та редагування інформації про мешканців гуртожитків;
- перевірка рейтингових балів;
- розселення мешканців по кімнатах;
- експорт інформації про дійсне розселення до Excel-файлу.

2. ВИБІР МЕТОДУ РІШЕННЯ

2.1 Вибір технологій

У даному пункті необхідно обґрунтувати два ключові моменти: причини, чому довелося відмовитися від інших популярних стеків технологій, та причини використання мінімуму технологій (наприклад, відсутність фреймворків). Звісно, обраний стек (PHP + JavaScript + jQuery + MySQL) може здатися дуже простим, застарілим, проте дану систему у майбутньому будуть підтримувати інші адміністратори і мови PHP та Javascript будуть більш універсальними ніж, наприклад, Ruby on Rails та Vue3. Виконаємо огляд мов програмування для бекенду та обраний тип бази даних.

Python, Node.js, Ruby, Java, C# – це найпопулярніші мови, що використовуються у веб-розробці [1], але жоден з них не є автентичним для веб-розробки, як PHP.

Node.js оптимальний фреймворк, особливо завдяки асинхронному підходу на заміну багатопоточності, що дозволяє уникнути стану «поточкового голоду», коли ресурсів вистачає лише на переключення між потоками та перевіркою їх стану [5]. Але система, яка проектується у нашому випадку, досить невелика і тому таких моментів не виникне.

Python та Ruby також інтерпретовані, динамічно типізовані мови, але PHP (особливо у версіях 7.x, 8.x) також зазнав значних оновлень. Але при всіх перевагах і простоті Python та Ruby дуже і дуже сильно поступаються PHP у веб-розробці [1].

2.2 Опис технологій для клієнтської частини

Беззаперечним лідером у розробці фронтенд складової є Javascript, оскільки і конкурентів-то фактично немає: Dart поки що підтримується лише в Dartium, ActionScript, який раніше хоча б якось але був задіяний, офіційно вже не використовується. TypeScript, CoffeeScript та всі інші подібні технології

переводяться в JavaScript і не є окремими самостійними мовами програмування [4].

Разом з JavaScript, були, звісно ж, використані html та css, бо разом вони є основою фронтенду [9]. Але крім цих трьох технологій була використана додатково бібліотека jQuery.

Багато хто може підняти питання про її актуальність, адже багато чого в JS уже є «з коробки» (під впливом, до речі, того ж jQuery [4]), є фреймворки, які створюють віртуальний DOM, а не модифікують. Є Vabel, який переводить код у новому стандарті в такий, що підтримується і в Internet Explorer 6 [9].

2.3 Опис технологій для серверної частини

У даному пункті варто зупинитися на двох основних речах: мова програмування (PHP) та тип БД (реляційна).

PHP було створено у 1994 році просто як набір CGI-скриптів, написаних мовою програмування C, що згадувалися під назвою “PHP (personal homepages) Tools” [2].

PHP 3.0 був першою версією, що нагадує сучасний PHP, тобто такий, яким ми знаємо його сьогодні. Одною із найсильніших сторін даної версії була можливість розширення ядра [2].

Версія PHP 4.0 уже було створено на базі власного рушія “Zend Engine” та мала в собі уже безліч функціоналу (наприклад, підтримка HTTP-сесій, буферизація виведення та нові конструкції) [1].

Станом на сьогодні, PHP має безліч переваг (починаючи з версії 5.x, але особливо після 7.x та 8.x) [1]:

- висока швидкість роботи та, відповідно, загальна продуктивність ресурсів;
- простота освоєння, простий синтаксис;
- висока гнучкість та функціональність (надзвичайно багата стандартна бібліотека);

- PHP був створений саме для веб-додатків (хоча і може використовуватися навіть в десктопі), що тягне за собою купу рішень, матеріалів та стандартного функціоналу саме для вебу;

- Можливість додавання PHP в html-код, що, звісно, не найкраща практика, але при правильному використанні в малих проектах є досить зручним та гарним рішенням (також можна згадати JSP-технологію, яка, на жаль, уже застаріла) [2].

Також під час розробки була використана саме реляційна база даних через те що, дані, що зберігаються та обробляються в нашій системі для автоматизації розселення студентів по гуртожиткам, є чітко структурованими та мають певні взаємозв'язки, що робить NoSQL бази даних не найкращим вибором в даній ситуації [7].

2.4 Інші використані технології

Замість великих та перенасичених IDE було використано Visual Studio Code (текстовий редактор з відкритим кодом на Electron.js, створений компанією Microsoft), що має ряд переваг перед тими ж Notepad++, Sublime Text, Atom та іншими, наприклад:

- зручний дебагер;
- вбудований термінал;
- зручні інструменти для роботи з git;
- підсвічування синтаксису для багатьох популярних мов та файлових форматів;
- зручна навігація;
- вбудований попередній перегляд Markdown;
- розумне авто доповнення;
- вбудований пакетний менеджер.

Також Visual Studio Code має велику кількість плагінів для розробника [5].

Для комфортної розробки було встановлено наступні плагіни:

- PHP Intelephense. У редакторі уже є підтримка синтаксису та підказки стандартних функцій мови PHP, але без спеціального доповнення редактор не буде підказувати власні функції розробника з інших частин проекту. Тому для підтримки автодоповнення, переходу до місця оголошення функції, класу чи змінної, аналізу коду і використовується даний пагін [5];

- PHP Debug. Під час розробки може виникнути ситуація, коли простих функцій логування та відладки уже недостатньо. Тоді може допомогти спеціальний інструмент – Дебагер. Для PHP є розширення xdebug, яка дозволяє ставити точки зупинки та переглянути оточення в місці, де, ймовірно, є помилка, виконуючи код поетапно або до наступної точки зупинки;

- Emmet. Допомагає пришвидшити верстання [2];

- Live Server. Щоб відразу бачити результати роботи над фронтенд частиною, потрібно відкрити файл за допомогою Live Server, що дасть змогу спостерігати за змінами після кожного збереження [2];

- REST Client. Своєрідна заміна Postman.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Створення бази даних

У якості бази даних була використана реляційна база даних – MariaDB. Її встановлення було легким завдяки XAMPP – безкоштовна збірка веб-сервера, що містить HTTP-сервер (Apache), систему керування базами даних MySQL, інтерпретатори для мов PHP/Perl, та, власне, базу даних MariaDB [5].

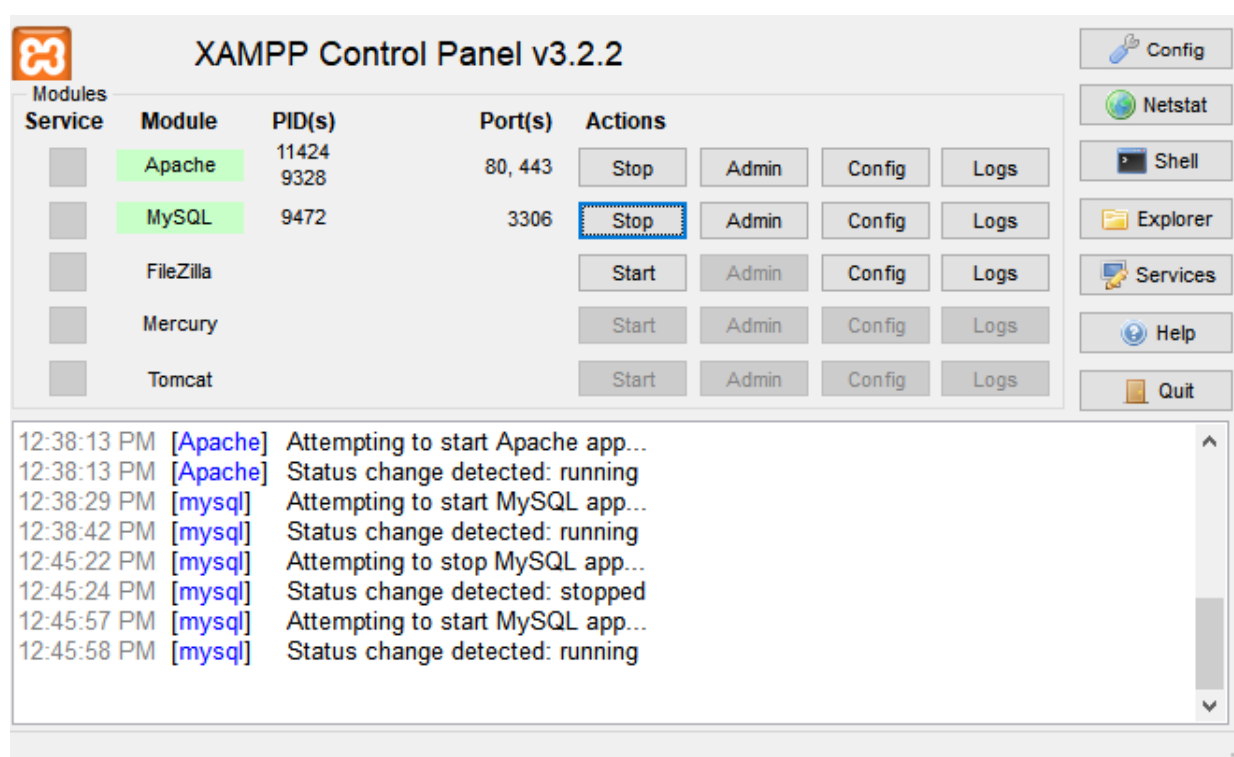


Рисунок 3.1 – Зовнішній вигляд та складові XAMPP

Приклад типових SQL-запитів, що використовувалися при створенні бази даних:

Створення таблиці з гуртожитками:

```

CREATE TABLE `hostels` (
  `ID` int(11) NOT NULL,
  `Title` varchar(100) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL,
  `Address` varchar(250) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL
)
  
```

Додавання тестових мешканців гуртожитку до бази:

```
INSERT INTO `Students` (`ID`, `Name`, `RoomID`, `SectionID`, `HostelID`, `Image`, `University`, `Speciality`, `rating`, `login`, `pass`) VALUES
(1, 'Вася Пупкин', 9, 0, 0, '', 'Название университета', 'Специальность', 51, '', ''),
(2, 'Веня Сечкин', 0, 0, 0, '', 'Название университета', 'Специальность', 41, '', ''),
(17, 'sdfsdfsdf', 9, 0, 0, '/students/visamc.png', 'sdfsdfsdf', 'sdfsdfsdf', 221, '', ''),
(40, 'qwe', 11, 0, 0, '/students/IMG_0985.JPG', 'qwe', 'qwe', 123, 'qwe', 'password');
```

Так, на жаль, зображення зберігаються на сервері, що негативно впливає на кількість вільного місця. Більш оптимальним рішенням було б зберігати в базі даних діючі посилання на фотографії, які, в свою чергу, знаходились би в якомусь cloud-сховищі.

3.2 Реалізація клієнтської та серверної частини

Насамперед, структура проекту:

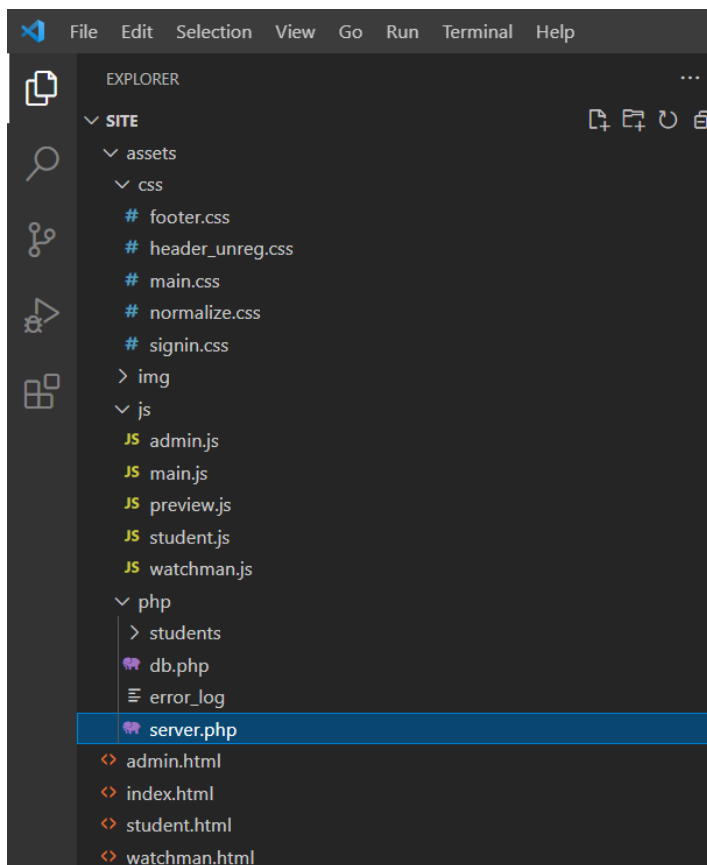


Рисунок 3.2 – Структура проекту

Система складається лише з чотирьох HTML-сторінок, оскільки дані підвантажуються динамічно завдяки технології AJAX [3]. Наприклад, підвантаження списку гуртожитків:

```
$(window).on('load', function () {
$.ajax({
url: '/assets/php/server.php',
type: 'POST',
data: { getHostels: "1" },
})
.done(function (data) {
showHostels(JSON.parse(data));
})
})
```

Як можна бачити, основна логіка знаходиться у файлі server.php.

Структури клієнтської частини:

- admin.html – “адмінка” системи;
- index.html – точка входу до сайту (логін сторінка);
- student.html – профіль студента;
- watchman.html – профіль вахтера.

Наприклад, запит на реєстрацію реалізовано завдяки тонкощам звичайного тегу <a>. Подібний підхід дозволяє відправити email в браузері або відкрити відповідний додаток на комп’ютері.

```
<header class="header">
  <a class="header__link" href="#">
    <div class="header__logo">MOOD 

```

Серед CSS-стилів наявний файл normalize.css, що є стороннім рішенням. Застосування даної бібліотеки дозволило нормалізувати структуру елементів та уніфікувати її в залежності від браузера та його версії.

Уся основна логіка бекенду міститься у файлі `server.php`. Але для зберігання інформації для коректного доступу до бази даних використовується файл `db.php`. Зразок для розгортання на `localhost`:

```
$host = "localhost";
$database = "mooddb";
$user = "mooddb";
і т.д.
```

Замість, наприклад, JWT-аутентифікації було використано більш простий підхід – використання HTTP-сесії. На початку роботи скрипта `server.php` сесія запускається командою `session_start()` [1]. Далі вже у залежності від вхідних даних, виконується відповідна логіка:

```
if (isset($_POST['exit'])) {
    session_unset();
}
if (isset($_POST['getCurrUser'])) {
    //логіка отримання діючого юзера
}
if (isset($_POST['auth'])) {
    //логіка автентифікації
}

if (isset($_POST['getHostels'])) {
    //логіка отримання списку гуртожитків
}

if (isset($_POST['CreateNewHostel'])) {
    //логіка створення гуртожитків
}

і т.д.
```

3.3 Інструкція користувача для адміністратора

Варіанти використання системи адміністратором наступні:

- редагування інформації;
- перегляд переліку студентів без кімнати;

- додавання студента до системи;
- розселення.

Звісно ж, для початку варто залогінитися. До речі, дані адміністратора вносяться до бази даних мануально. Далі до уваги адміністратора є декілька вкладок: “Редагування інформації” та “Розселення студентів”.

“Редагування інформації” дозволяє додавати гуртожитки, секції та кімнати в інтерактивному режимі:

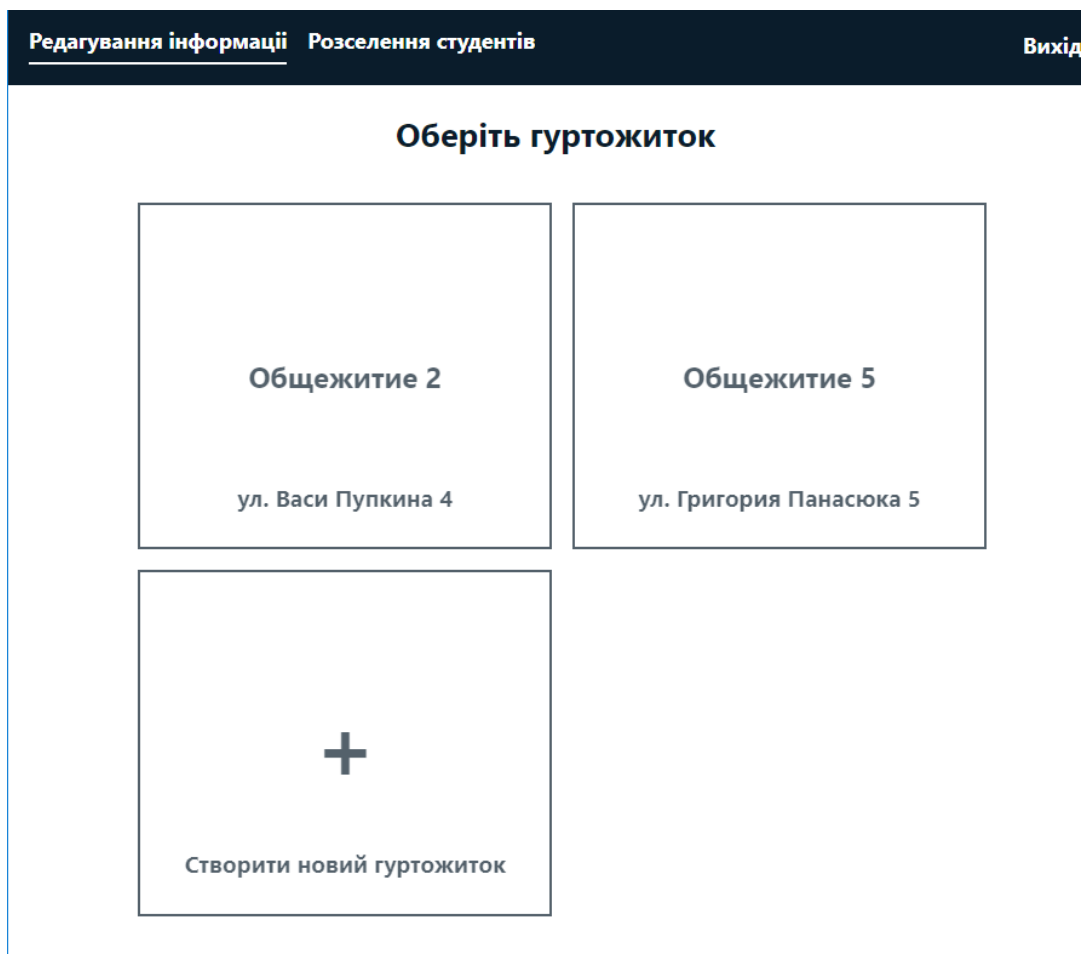


Рисунок 3.3 – «Редагування інформації» адміністратором

Оберіть гуртожиток

<p>Тест</p> <p>ТестТест</p>	<p>Общежитие 2</p> <p>ул. Васи Пупкина 4</p>
<p>Общежитие 5</p> <p>ул. Григория Панасюка 5</p>	<p>+</p> <p>Створити новий гуртожиток</p>

Рисунок 3.4 – Додавання нового гуртожитку

Для додавання секції чи кімнати процес виглядає аналогічно.

Вкладка “Розселення студентів” складається з трьох видимих частин та однієї “невидимої” – при натисненні комбінації CTRL-S можна зберегти дані про розселених студентів в форматі Excel.



Рисунок 3.5 – Секції вкладки “Розселення студентів”

Вкладка “Розселення” (активна за замовчуванням) є дуже подібною за функціоналом до “Редагування інформації” (описана вище), але дозволяє додати студента до кімнати:

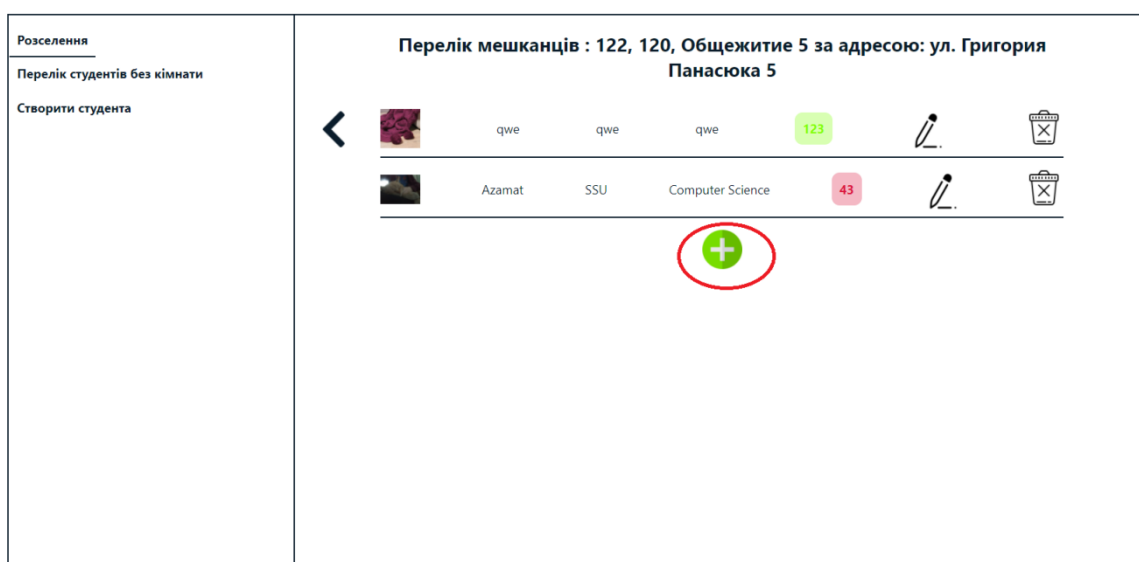


Рисунок 3.6 – Натиснення кнопки для додавання мешканця

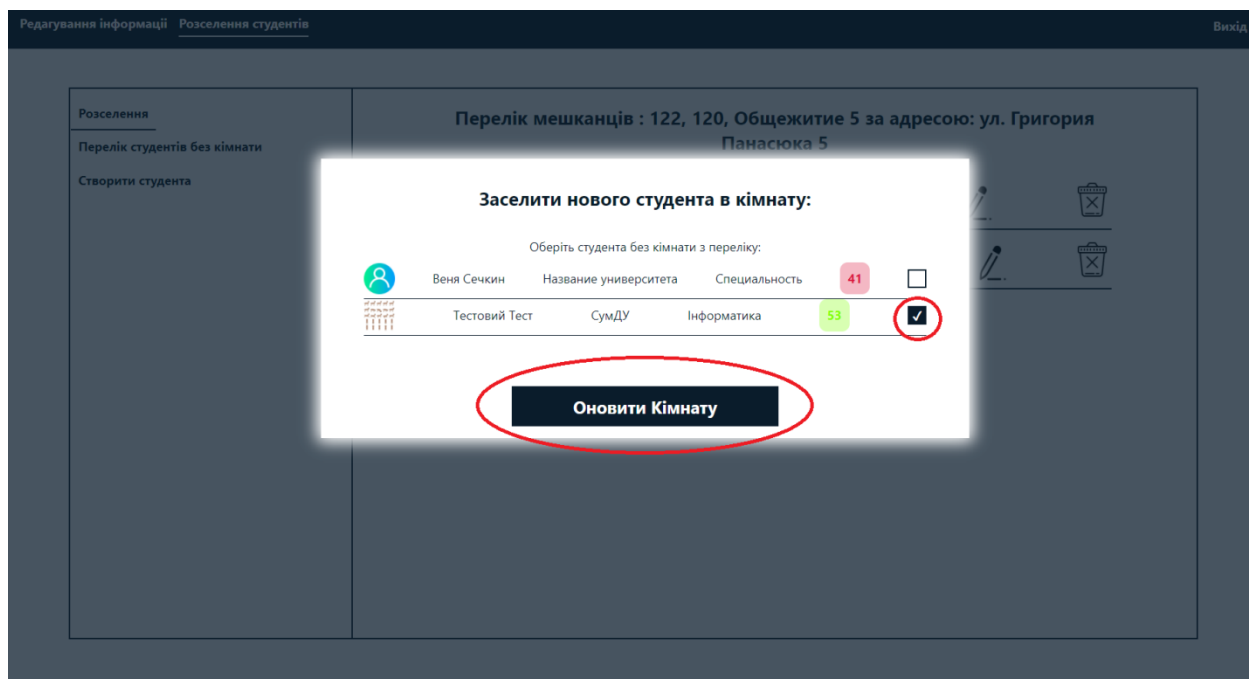


Рисунок 3.7 – Вибір нового мешканця зі списку

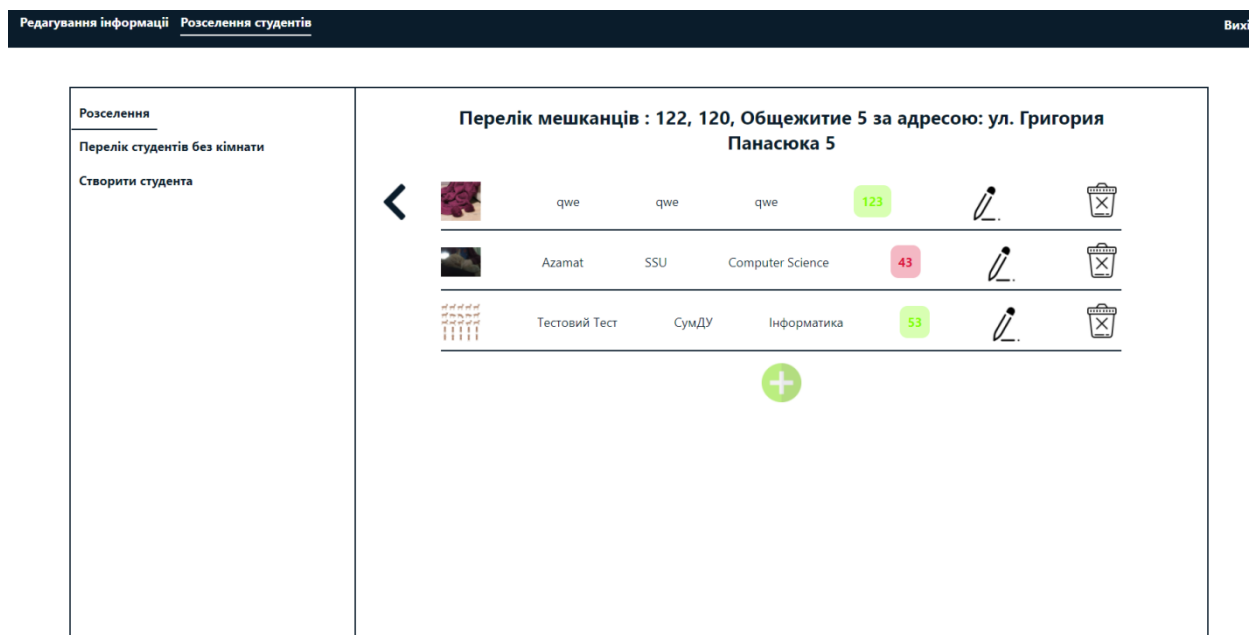


Рисунок 3.8 – Оновлений список мешканців

Вкладка “Перелік студентів без кімнати” дозволяє переглянути список нерозселених студентів (до речі, рейтинговий бал змінює колір в залежності від того, чи достатньо набрано балів для поселення):

Розселення	Перелік студентів без кімнати					
Перелік студентів без кімнати						
Створити студента						
		Веня Сечкин	Название университета	Специальность	41	
		Студент	СумДУ	Переклад	55	

Рисунок 3.9 – Список не розселених студентів

Вкладка “Створити студента” дозволяє додати нового студента до бази даних:

Розселення	Створення нового студента
Перелік студентів без кімнати	
Створити студента	
	<p>Повне ім'я</p> <input type="text" value="Створений Студент Ааааа"/>
	<p>Назва університету</p> <input type="text" value="СумДУ"/>
	<p>Спеціальність</p> <input type="text" value="Інформатика"/>
	<p>Рейтинг</p> <input type="text" value="69"/>
	<p>Логін</p> <input type="text" value="ssss2222ss"/>
	<p>Пароль</p> <input type="password" value="....."/>
	<p>Змінити</p> <input type="button" value="Виберіть файл"/> Файл не вибран
	<input type="button" value="Створити"/>

Рисунок 3.10 – Створення нового студента

3.4 Інструкція користувача для вахтера

Для вахтерів передбачено лише один шлях використання – це перевірка факту проживання студента в гуртожитку, але він виконується трьома способами: пошук за кімнатою, пошук за прізвищем та вибір ієрархії гуртожиток ^{s_u} секція ^{s_u} кімната зі списку.

Пошук за кімнатою:

1. Перейти на сторінку логіна:

MOOD 🤖 Contact to sign up

Login:

Password:

Developed by MacGregory-is-codding
2021

Рисунок 3.11 – Логін сторінка

2. Ввести раніше отримані від адміністратора чи відповідальної за це особи реквізити та натиснути кнопку входу:

Login:

Password:

Sign In

Рисунок 3.12 – Введення реквізитів

3. Ввести в поле пошуку номер кімнати, заявлений студентом, та звітритися з результатами:

Вихід

Пошук

Оберіть гуртожиток

Общежитие 2 ул. Васи Пупкина 4	Общежитие 5 ул. Григория Панасюка 5
-----------------------------------	--

Рисунок 3.13 – Введення кімнати для пошуку

Вихід

Пошук

Результати пошуку за запитом - 122 :

122, 120 , Общежитие 5 за адресою - ул. Григория Панасюка 5



	Тестовий Студент Тест 123
	Azamat 43

X

Рисунок 3.14 – Результати пошуку

Пошук за прізвищем (або іншою частиною імені):

1. Перейти на сторінку логіна:

MOOD  Contact to sign up 



Login:

Password:

Developed by MacGregory-is-codding
2021

Рисунок 3.15 – Логін сторінка

2. Ввести раніше отримані від адміністратора чи відповідальної за це особи реквізити та натиснути кнопку входу:

MOOD  Contact to sign up 

Login:

Password:

Developed by MacGregory-is-codding
2021

Рисунок 3.16 – Введення реквізитів

3. Ввести в поле пошуку прізвище, заявлене студентом, та звіритися з результатами:

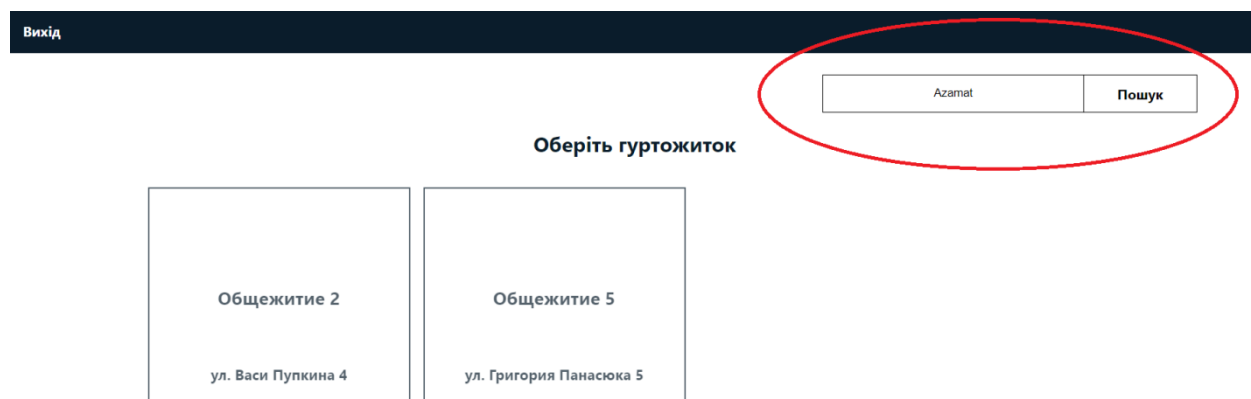


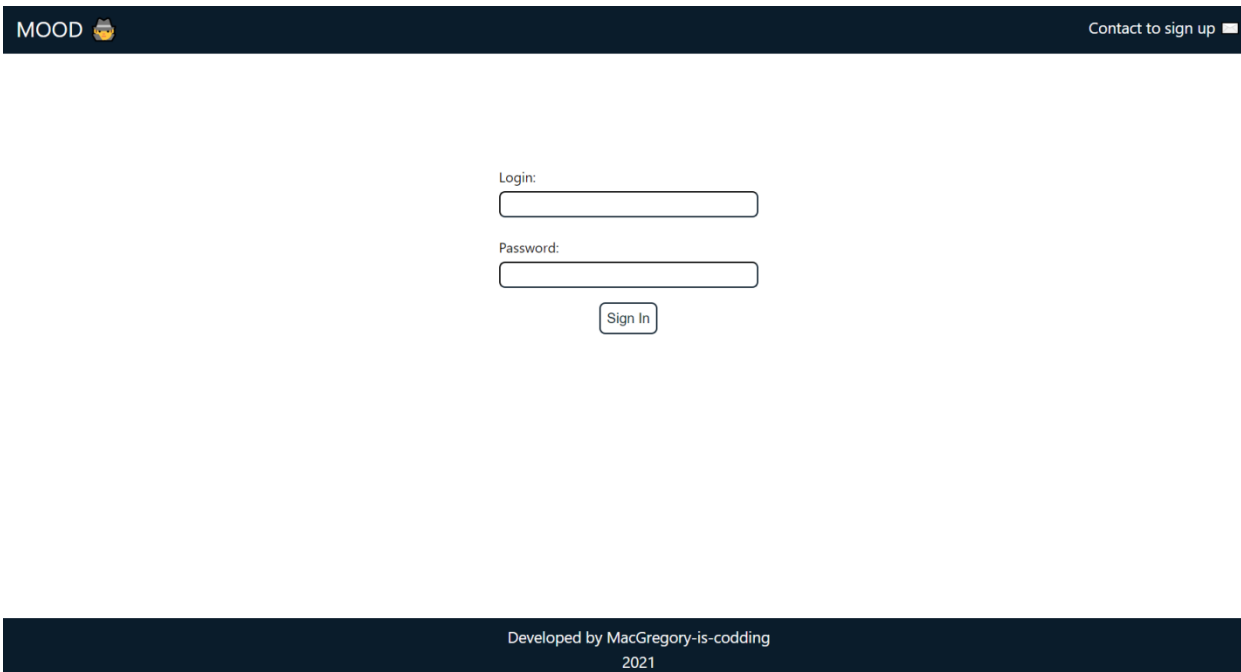
Рисунок 3.17 – Введення прізвища



Рисунок 3.18 – Результати пошуку за прізвищем

Пошук з вибором ієрархії:

1. Перейти на сторінку логіна:



MOOD 🐻 Contact to sign up ▢

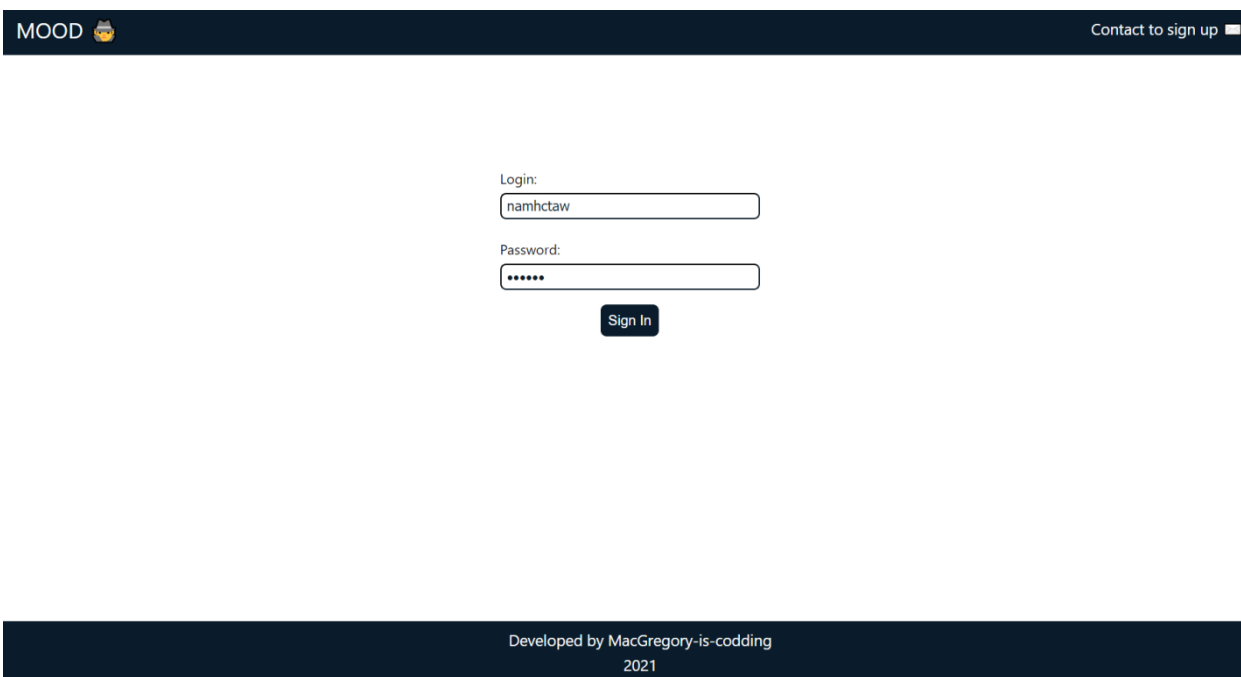
Login:

Password:

Developed by MacGregory-is-codding
2021

Рисунок 3.19 – Логін сторінка

2. Ввести раніше отримані від адміністратора чи відповідальної за це особи реквізити та натиснути кнопку входу:



MOOD 🐻 Contact to sign up ▢

Login:

Password:

Developed by MacGregory-is-codding
2021

Рисунок 3.20 – Введення реквізитів

3. Вибір відповідної ієрархії, наприклад:

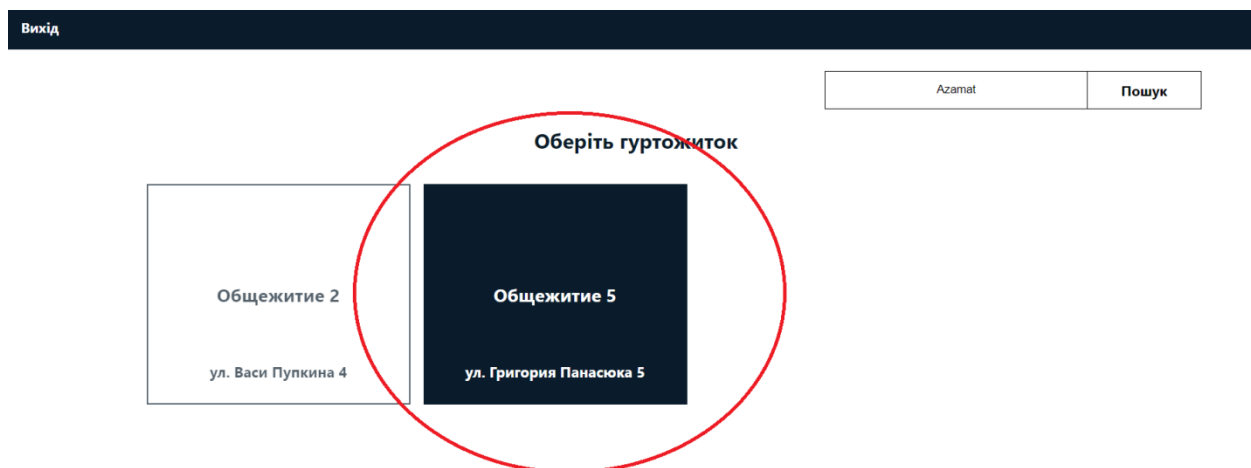


Рисунок 3.21 – Вибір гуртожитку з ієрархії

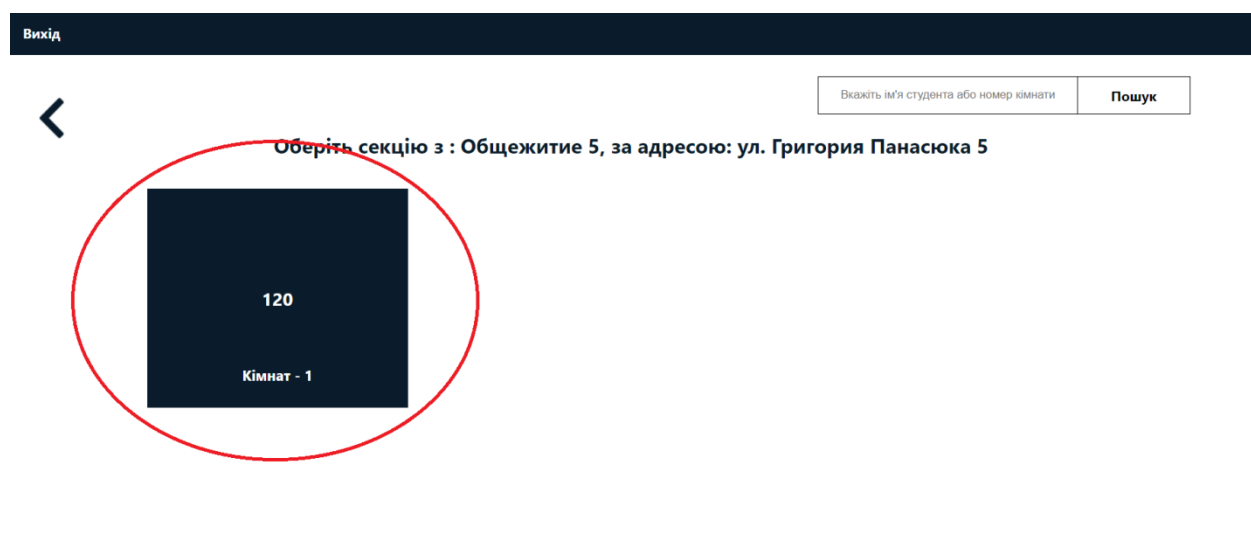


Рисунок 3.22 – Вибір секції з ієрархії

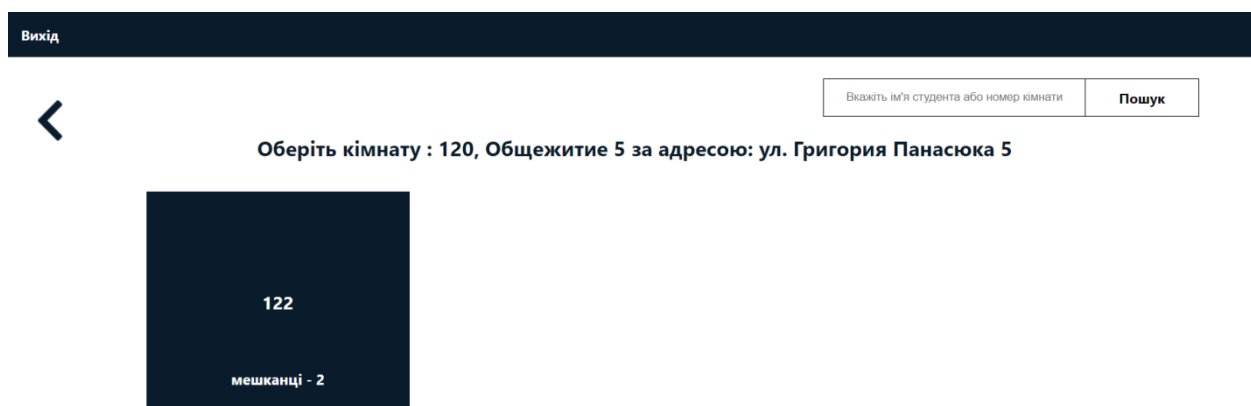


Рисунок 3.23 – Вибір кімнати з ієрархії

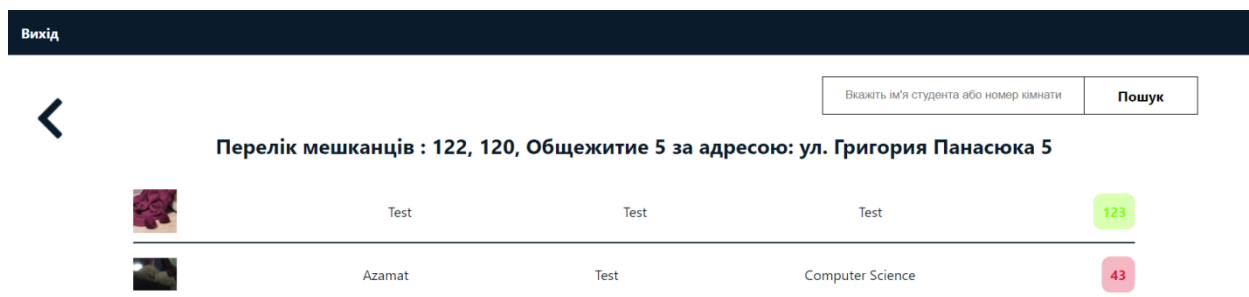


Рисунок 3.24 – Список мешканців, що проживають в обраній кімнаті

3.5 Інструкція користувача для студента

У студента є декілька варіантів використання системи, а саме перевірка персональної інформації та запит на реєстрацію в системі.

Перевірка персональної інформації:

1. Перейти на сайт

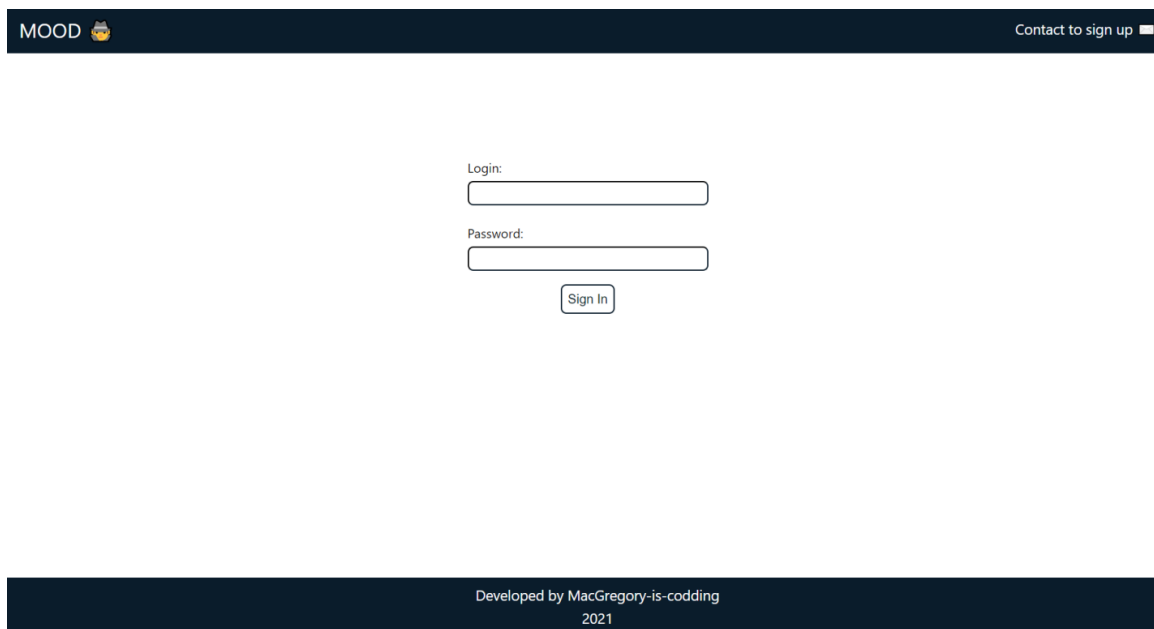




Рисунок 3.25 – Логін сторінка

2. Ввести логін та пароль, що були попередньо видані адміністратором чи відповідальною за це особою

MOOD  Contact to sign up 

Login:

Password:

Developed by MacGregory-is-coding
2021

Рисунок 3.26 – Введення реквізитів

3. Результат:



Вихід

Тестовий Тест
 СумДУ
 Інформатика
53
 test
 Студент без кімнати


Рисунок 3.27 – Зразок профіля мешканця гуртожитку


Надсилення запиту про реєстрацію.

1. Натиснути відповідну кнопку, що знаходиться на сторінці логіну:

MOOD  Contact to sign up 

Как вы хотите открыть это?

 Google Chrome
Создать

 Microsoft Outlook
Создать

Всегда использовать это приложение

Рисунок 3.28 – Надсилення запиту про реєстрацію

Отже, як бачимо з конкретних ілюстрацій та шляхів використання, розроблений продукт є інтуїтивно зрозумілим, простим та зручним у використанні та відповідає заявленому функціоналу.

ВИСНОВКИ

У ході виконання випускної роботи було розглянуто питання автоматизації розселення студентів по гуртожиткам та засоби веб-розробки. Також було проведено аналіз тематичної літератури, досліджена доцільність та актуальність роботи.

Результатом виконання даної роботи є створений веб-сайт для автоматизації розселення студентів по гуртожиткам із використанням технологій PHP, JavaScript, MySQL та jQuery.

У результаті роботи було виконане поставлене завдання, а саме розробка системи для автоматизації розселення студентів по гуртожиткам.

Розроблена система включає в себе:

- спроектовану базу даних та інтерфейси для її заповнення (адмін. панель);
- різні користувацькі ролі;
- функціонал розселення та звітності по студентам;
- функціонал перевірки факту проживання студента в гуртожитку вахтером та перегляд особистої інформації та рейтингового балу студентом;
- адаптивний дизайн сторінок для студентів та вахтерів, що дозволяє користуватися веб-додатком на різних пристроях.

Подібного роду системи є досить складними і в плані проектування, і в плані розробки. При їх створенні варто враховувати та аналізувати цілий комплекс чинників, але насамперед – хто та як буде використовувати дану систему. Розроблений сайт має зручний та інтуїтивний функціонал та задовольняє всім вимогам, які ставилися на етапі постановки завдання.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Бардзелл Джеффри Macromedia Dreamweaver MX 2004 с ASP, ColdFusion и PHP. Из первых рук (+ CD-ROM); Эком - М., **2016**. - 560 с.
2. Бенкен Елена PHP, MySQL, XML. Программирование для Интернета; БХВ-Петербург - М., **2017**. - 336 с.
3. Дронов Владимир JavaScript и AJAX в Web-дизайне; БХВ-Петербург - М., **2015**. - 736 с.
4. Бибо Бер , Кац Иегуда jQuery. Подробное руководство по продвинутому JavaScript; Символ-плюс - М., **2017**. - 624 с.
5. Клименко Роман Веб-мастеринг на 100%; Питер - М., 2015. - **920** с.
6. Роббинс Дженнифер HTML5, CSS3 и JavaScript. Исчерпывающее руководство (+ DVD-ROM); Эксмо - М., **2017**. - 528 с.
7. Joseph, J. Vambara SQL Server® Developer's Guide / Joseph J. Vambara, Paul R. Allen. - Москва: **Мир**, **2016**. - **235** с.
8. Грабер, Мартин SQL для простых смертных / Мартин Грабер. - М.: ЛОРИ, 2014. - 378 с.
9. Andy, Harris HTML, XHTML and CSS All-In-One For Dummies® / Andy Harris. - Москва: **Наука**, **2014**. - **173** с.

ДОДАТОК А – ЛІСТИНГ ПРОГРАМИ

- admin.html

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="/assets/css/main.css">
  <title>Панель Адміністратора</title>
</head>

<body>
  <header>
    <ul class="menu">
      <li class="menu_item menu_item_active" menu-type="db">Редагування інформації</li>
      <li class="menu_item" menu-type="cr">Розселення студентів</li>
    </ul>
    <div class="exit">
      Вихід
    </div>
  </header>
  <section class="edit_db">
    
    <section class="db_res_wrap hostels">
      <h2>Оберіть гуртожиток</h2>
      <div class="db_res_list">
      </div>
      <div class="create_new create_new_hostel">
        <div class="create_new_body">
          <h2>Створення нового гуртожитка:</h2>
          <div class="input_wrap">
            <label>Вкажіть назву гуртожитка</label>
            <input type="text" class="new_hostel_title" />
          </div>
          <div class="input_wrap">
            <label>Вкажіть адресу гуртожитка:</label>
            <input type="text" class="new_hostel_address" />
          </div>
          <div class="create_hostel create_new_btn">Створити</div>
        </div>
      </div>
    </section>
    <section class="db_res_wrap sections">
      <h2></h2>
      <div class="db_res_list">
      </div>
      <div class="create_new create_new_section">
        <div class="create_new_body">
          <h2>Створення нової секції:</h2>
          <div class="input_wrap">
            <label>Вкажіть назву секції</label>
            <input type="text" class="new_section_title" />
          </div>
          <div class="create_section create_new_btn">Створити</div>
        </div>
      </div>
    </section>
  </body>

```

```

    </div>
  </div>
</section>
<section class="db_res_wrap rooms">
  <h2></h2>
  <div class="db_res_list">
  </div>
  <div class="create_new create_new_room">
    <div class="create_new_body">
      <h2>Створення нової кімнати:</h2>
      <div class="input_wrap">
        <label>Вкажіть номер кімнати</label>
        <input type="text" class="new_room_title" />
      </div>
      <div class="create_room create_new_btn">Створити</div>
    </div>
  </div>
</section>
<section class="db_res_wrap students">
  <h2></h2>
  <div class="users_list">
  </div>
</section>
</section>
<section class="change_room">
  <div class="change_room_body">
    <div class="select_event">
      <div class="event_change_room selected_event">Розселення</div>
      <div class="event_list_users_without_room">Перелік студентів без кімнати</div>
      <div class="event_create_user">Створити студента</div>
    </div>
    <div class="select_event_body">
      <div class="change_room_block">
        
        <section class="db_res_wrap hostels">
          <h2>Оберіть гуртожиток</h2>
          <div class="db_res_list">
          </div>
          <div class="create_new create_new_hostel">
            <div class="create_new_body">
              <h2>Створення нового гуртожитка:</h2>
              <div class="input_wrap">
                <label>Вкажіть назву гуртожитка</label>
                <input type="text" class="new_hostel_title" />
              </div>
              <div class="input_wrap">
                <label>Вкажіть адресу гуртожитка:</label>
                <input type="text" class="new_hostel_address" />
              </div>
              <div class="create_hostel create_new_btn">Створити</div>
            </div>
          </div>
        </section>
        <section class="db_res_wrap sections">
          <h2></h2>
          <div class="db_res_list">
          </div>
          <div class="create_new create_new_section">
            <div class="create_new_body">

```

```

    <h2>Створення нової секції:</h2>
    <div class="input_wrap">
      <label>Вкажіть назву секції</label>
      <input type="text" class="new_section_title" />
    </div>
    <div class="create_section create_new_btn">Створити</div>
  </div>
</div>
</section>
<section class="db_res_wrap rooms">
  <h2></h2>
  <div class="db_res_list">
  </div>
  <div class="create_new create_new_room">
    <div class="create_new_body">
      <h2>Створення нової кімнати:</h2>
      <div class="input_wrap">
        <label>Вкажіть номер кімнати</label>
        <input type="text" class="new_room_title" />
      </div>
      <div class="create_room create_new_btn">Створити</div>
    </div>
  </div>
</section>
<section class="db_res_wrap students">
  <h2></h2>
  <div class="users_list">
  </div>
</section>
</div>
<div class="list_users_without_room_body">
  <h2>Перелік студентів без кімнати</h2>
  <div class="list_users_without_room_body_list">

  </div>
</div>
<div class="create_user_body">
  <h2>Створення нового студента</h2>
  <div class="input_wrap">
    <label>Повне ім'я</label>
    <input type="text" class="new_user_name">
  </div>
  <div class="input_wrap">
    <label>Назва університету</label>
    <input type="text" class="new_user_univ">
  </div>
  <div class="input_wrap">
    <label>Спеціальність</label>
    <input type="text" class="new_user_spec">
  </div>
  <div class="input_wrap">
    <label>Рейтинг</label>
    <input type="text" class="new_user_rating">
  </div>
  <div class="input_wrap">
    <label>Логін</label>
    <input type="text" class="new_user_login">
  </div>
  <div class="input_wrap">

```

```

        <label>Пароль</label>
        <input type="password" class="new_user_pass">
    </div>
    <div class="input_wrap">
        <label class="new_user_image_label">Зображення</label>
        <input type="file" class="new_user_image">
        <div class="image_preview"></div>
    </div>
    <div class="new_user_btn create_student">
        Створити
    </div>
</div>
</div>
</div>
</section>
<script src="https://code.jquery.com/jquery-3.6.0.js"
    integrity="sha256-H+K7U5CnXl1h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk=" crossorigin="anonymous">
</script>
<script type="text/javascript" src="assets/js/preview.js"></script>

<script src="/assets/js/admin.js"></script>
</body>

</html>

```

- **index.html**

```

<!DOCTYPE html>
<html>

<head lang="en">
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Sign In</title>
    <link rel="stylesheet" href="/assets/css/normalize.css">
    <link rel="stylesheet" href="/assets/css/signin.css">
    <link rel="stylesheet" href="/assets/css/header_unreg.css">
    <link rel="stylesheet" href="/assets/css/footer.css">
</head>

<body>
    <header class="header">
        <a class="header__link" href="#">
            <div class="header__logo">MOOD 

```

```

        id="password"></section>
    <section class="signin-form__submit-section"><input class="signin-form__submit-button" type="submit"
        value="Sign In"></section>
    </form>
</div>
</main>
<footer class="footer">
    <span class="footer__span">Developed by MacGregory-is-codding</span>
    <span class="footer__span">2021</span>
</footer>
<script src="https://code.jquery.com/jquery-3.6.0.js" integrity="sha256-
H+K7U5CnXl1h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk=" crossorigin="anonymous">
</script>
<script src="/assets/js/main.js"></script>
</body>
</html>

```

- student.html

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/assets/css/main.css">
    <title>Панель Студента</title>
</head>

<body>
    <header>
        <div class="exit">
            Вихід
        </div>
    </header>
    <section class="student_page">

</section>
<script src="https://code.jquery.com/jquery-3.6.0.js"
    integrity="sha256-H+K7U5CnXl1h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk=" crossorigin="anonymous">
    </script>

    <script src="/assets/js/student.js"></script>
</body>

</html>

```

- watchman.html

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="/assets/css/main.css">
  <title>Панель Вахтера</title>
</head>

<body>
  <header>
    <div class="exit">
      Вихід
    </div>
  </header>
  <div class="search">
    <input type="text" class="search_input" placeholder="Вкажіть ім'я студента або номер кімнати"/>
    <div class="search_btn">Пошук</div>
  </div>
  <section class="edit_db">
    
    <section class="db_res_wrap hostels">
      <h2>Оберіть гуртожиток</h2>
      <div class="db_res_list">
      </div>
    </section>
    <section class="db_res_wrap sections">
      <h2></h2>
      <div class="db_res_list">
      </div>
    </section>
    <section class="db_res_wrap rooms">
      <h2></h2>
      <div class="db_res_list">
      </div>
    </section>
    <section class="db_res_wrap students">
      <h2></h2>
      <div class="users_list">
      </div>
    </section>
  </section>
  <div class="search_result">
    <h2></h2>
    
    <div class="search_result_body">

    </div>
  </div>

  <script src="https://code.jquery.com/jquery-3.6.0.js"
    integrity="sha256-H+K7U5CnXl1h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk=" crossorigin="anonymous">
  </script>
  <script type="text/javascript" src="assets/js/preview.js"></script>

  <script src="/assets/js/watchman.js"></script>
</body>

</html>

```

- db.php

```
<?php
$host = "localhost"; // хост
$database = "stechlea_community"; // имя базы данных
$user = "stechlea"; // пользователь
$password = "G@hjkm192168"; // пароль
$connect = mysqli_connect($host, $user, $password, $database); //подключение к БД
mysqli_set_charset($connect, "utf8");
mysqli_query($connect, "SET NAMES utf8");

//-----Этот файл подключен к всем .php файлам которые работаю с БД
?>
```

- server.php

```
<?
session_start();
//$_SESSION['User'] = array('type' => 'student', 'ID' => 1);
require_once('db.php');

if (isset($_POST['exit'])) {
    session_unset();
}
if (isset($_POST['getCurrUser'])) {
    if (isset($_SESSION['User'])) {
        echo json_encode($_SESSION['User']);
    } else {
        echo 0;
    }
}
if (isset($_POST['auth'])) {
    $login = $_POST['login'];
    $pass = $_POST['pass'];
    //echo $pass;
    $res = -1;
    if ($login == 'namhctaw' && $pass == 'Gfhjkm') {
        $_SESSION['User'] = array('type' => 'watchman');
        $res = 0;
    }
    if ($login == 'nimda' && $pass == '9379992') {
        $_SESSION['User'] = array('type' => 'admin');
        $res = 1;
    }
}
//$hashPass = password_hash($pass, PASSWORD_DEFAULT);
$q = mysqli_query($connect, "SELECT ID FROM Students WHERE login='$login' AND pass='$pass'");
echo mysqli_error($connect);
$row_cnt = mysqli_num_rows($q);
echo $hashPass;
if ($row_cnt > 0) {
    $res = 2;
    while ($r = mysqli_fetch_assoc($q)) {
        $_SESSION['User'] = array('type' => 'student', 'ID' => $r['ID']);
        $res = 2;
    }
}
```

```

    }
  }
  echo $res;
}

if (isset($_POST['getHostels'])) {
  $Hostels = array();
  $q = mysqli_query($connect, "SELECT ID,Title,Address FROM hostels");
  while ($r = mysqli_fetch_assoc($q)) {
    $Hostels[] = $r;
  }
  echo mysqli_error($connect);
  echo json_encode($Hostels);
}

if (isset($_POST['CreateNewHostel'])) {
  $title = $_POST['title'];
  $address = $_POST['address'];
  mysqli_query($connect, "INSERT INTO `hostels`(`Title`, `Address`) VALUES ('$title','$address')");
  $Hostels = array();
  $q = mysqli_query($connect, "SELECT ID,Title,Address FROM hostels");
  while ($r = mysqli_fetch_assoc($q)) {
    $Hostels[] = $r;
  }
  echo mysqli_error($connect);
  echo json_encode($Hostels);
}

if (isset($_POST['UpdateHostel'])) {
  $ID = $_POST['id'];
  $title = $_POST['title'];
  $address = $_POST['address'];
  mysqli_query($connect, "UPDATE `hostels` SET Title='$title', Address='$address' WHERE ID=$ID");
  $Hostels = array();
  $q = mysqli_query($connect, "SELECT ID,Title,Address FROM hostels");
  while ($r = mysqli_fetch_assoc($q)) {
    $Hostels[] = $r;
  }
  echo mysqli_error($connect);
  echo json_encode($Hostels);
}

if (isset($_POST['DeleteHostel'])) {
  $id = $_POST['id'];
  mysqli_query($connect, "DELETE FROM `hostels` WHERE ID=$id");
  $Hostels = array();
  $q = mysqli_query($connect, "SELECT ID,Title,Address FROM hostels");
  while ($r = mysqli_fetch_assoc($q)) {
    $Hostels[] = $r;
  }
  echo mysqli_error($connect);
  echo json_encode($Hostels);
}

/*****Sections*****/
/*****/
if (isset($_POST['getSectionsByHostel'])) {
  $id = $_POST['id'];
  $Sections = array();
  $q = mysqli_query($connect, "SELECT ID,Title FROM Section WHERE HostelID=$id");
  while ($r = mysqli_fetch_assoc($q)) {
    $Sections[] = array('ID' => $r['ID'], 'Title' => $r['Title'], 'Count' => 0);
  }
}

```



```

}
for ($i = 0; $i < count($Sections); $i++) {
    $id = $Sections[$i]['ID'];
    $q = mysqli_query($connect, "SELECT ID FROM Room WHERE SectionID=$id");
    if ($result = mysqli_query($connect, "SELECT ID FROM Room WHERE SectionID=$id")) {
        $row_cnt = mysqli_num_rows($result);
        $Sections[$i]['Count'] = $row_cnt;
    }
}
echo mysqli_error($connect);
echo json_encode($Sections);
}
if (isset($_POST['UpdateSection'])) {
    $ID = $_POST['id'];
    $title = $_POST['title'];
    mysqli_query($connect, "UPDATE `Section` SET Title='$title' WHERE ID=$ID");
    $Sections = array();
    $HostelID = 0;
    $q = mysqli_query($connect, "SELECT HostelID FROM Section WHERE ID=$ID");
    while ($r = mysqli_fetch_assoc($q)) {
        $HostelID = $r['HostelID'];
    }
    $q = mysqli_query($connect, "SELECT ID,Title FROM Section WHERE HostelID=$HostelID");
    while ($r = mysqli_fetch_assoc($q)) {
        $Sections[] = array('ID' => $r['ID'], 'Title' => $r['Title'], 'Count' => 0);
    }
    for ($i = 0; $i < count($Sections); $i++) {
        $id = $Sections[$i]['ID'];
        $q = mysqli_query($connect, "SELECT ID FROM Room WHERE SectionID=$id");
        if ($result = mysqli_query($connect, "SELECT ID FROM Room WHERE SectionID=$id")) {
            $row_cnt = mysqli_num_rows($result);
            $Sections[$i]['Count'] = $row_cnt;
        }
    }
    echo mysqli_error($connect);
    echo json_encode($Sections);
}
if (isset($_POST['DeleteSection'])) {
    $id = $_POST['id'];
    $HostelID = 0;
    $Sections = array();
    $q = mysqli_query($connect, "SELECT HostelID FROM Section WHERE ID=$ID");
    while ($r = mysqli_fetch_assoc($q)) {
        $HostelID = $r['HostelID'];
    }
    mysqli_query($connect, "DELETE FROM `Section` WHERE ID=$id");
    mysqli_query($connect, "UPADTE `Room` SET SectionID=0 WHERE SectionID=$id");

    $q = mysqli_query($connect, "SELECT HostelID FROM Section WHERE ID=$ID");
    while ($r = mysqli_fetch_assoc($q)) {
        $HostelID = $r['HostelID'];
    }
    $q = mysqli_query($connect, "SELECT ID,Title FROM Section WHERE HostelID=$HostelID");
    while ($r = mysqli_fetch_assoc($q)) {
        $Sections[] = array('ID' => $r['ID'], 'Title' => $r['Title'], 'Count' => 0);
    }
    for ($i = 0; $i < count($Sections); $i++) {
        $id = $Sections[$i]['ID'];
        $q = mysqli_query($connect, "SELECT ID FROM Room WHERE SectionID=$id");

```

```

        if ($result = mysqli_query($connect, "SELECT ID FROM Room WHERE SectionID=$id")) {
            $row_cnt = mysqli_num_rows($result);
            $Sections[$i]['Count'] = $row_cnt;
        }
    }
    echo mysqli_error($connect);
    echo json_encode($Sections);
}
if (isset($_POST['CreateNewSection'])) {
    $title = $_POST['title'];
    $HostelID = $_POST['hostelID'];
    mysqli_query($connect, "INSERT INTO `Section`(`Title`,`HostelID`) VALUES ('$title',$HostelID)");
    $Sections = array();
    $q = mysqli_query($connect, "SELECT ID,Title FROM Section WHERE HostelID=$HostelID");
    while ($r = mysqli_fetch_assoc($q)) {
        $Sections[] = array('ID' => $r['ID'], 'Title' => $r['Title'], 'Count' => 0);
    }
    for ($i = 0; $i < count($Sections); $i++) {
        $id = $Sections[$i]['ID'];
        $q = mysqli_query($connect, "SELECT ID FROM Room WHERE SectionID=$id");
        if ($result = mysqli_query($connect, "SELECT ID FROM Room WHERE SectionID=$id")) {
            $row_cnt = mysqli_num_rows($result);
            $Sections[$i]['Count'] = $row_cnt;
        }
    }
    echo mysqli_error($connect);
    echo json_encode($Sections);
}

/*****ROOMS*****/
/*****/
if (isset($_POST['getRoomsBySection'])) {
    $sid = $_POST['sectionId'];
    $Rooms = array();
    $q = mysqli_query($connect, "SELECT ID,Title FROM Room WHERE SectionID=$sid");
    while ($r = mysqli_fetch_assoc($q)) {
        $Rooms[] = array('ID' => $r['ID'], 'Title' => $r['Title'], 'Count' => 0);
    }
    for ($i = 0; $i < count($Rooms); $i++) {
        $id = $Rooms[$i]['ID'];
        if ($result = mysqli_query($connect, "SELECT ID FROM Students WHERE RoomID=$id")) {
            $row_cnt = mysqli_num_rows($result);
            $Rooms[$i]['Count'] = $row_cnt;
        }
    }
    echo mysqli_error($connect);
    echo json_encode($Rooms);
}
if (isset($_POST['UpdateRoom'])) {
    $ID = $_POST['id'];
    $title = $_POST['title'];
    mysqli_query($connect, "UPDATE `Room`SET Title='$title' WHERE ID=$ID");
    $Rooms = array();
    $SectionID = 0;
    $q = mysqli_query($connect, "SELECT SectionID FROM Room WHERE ID=$ID");
    while ($r = mysqli_fetch_assoc($q)) {
        $SectionID = $r['SectionID'];
    }
    $q = mysqli_query($connect, "SELECT ID,Title FROM Room WHERE SectionID=$SectionID");

```

```

while ($r = mysqli_fetch_assoc($q)) {
    $Rooms[] = array('ID' => $r['ID'], 'Title' => $r['Title'], 'Count' => 0);
}
for ($i = 0; $i < count($Rooms); $i++) {
    $id = $Rooms[$i]['ID'];
    if ($result = mysqli_query($connect, "SELECT ID FROM Students WHERE RoomID=$id")) {
        $row_cnt = mysqli_num_rows($result);
        $Rooms[$i]['Count'] = $row_cnt;
    }
}
echo mysqli_error($connect);
echo json_encode($Rooms);
}
if (isset($_POST['DeleteRoom'])) {
    $id = $_POST['id'];
    $SectionID = 0;
    $Sections = array();
    $q = mysqli_query($connect, "SELECT SectionID FROM Room WHERE ID=$id");
    while ($r = mysqli_fetch_assoc($q)) {
        $SectionID = $r['SectionID'];
    }
    mysqli_query($connect, "DELETE FROM `Room` WHERE ID=$id");
    mysqli_query($connect, "UPADTE `Students` SET RoomID=0 WHERE RoomID=$id");
    $Rooms = array();
    $q = mysqli_query($connect, "SELECT ID,Title FROM Room WHERE SectionID=$SectionID");
    while ($r = mysqli_fetch_assoc($q)) {
        $Rooms[] = array('ID' => $r['ID'], 'Title' => $r['Title'], 'Count' => 0);
    }
    for ($i = 0; $i < count($Rooms); $i++) {
        $id = $Rooms[$i]['ID'];
        if ($result = mysqli_query($connect, "SELECT ID FROM Students WHERE RoomID=$id")) {
            $row_cnt = mysqli_num_rows($result);
            $Rooms[$i]['Count'] = $row_cnt;
        }
    }
    echo mysqli_error($connect);
    echo json_encode($Rooms);
}
if (isset($_POST['CreateNewRoom'])) {
    $title = $_POST['title'];
    $SectionID = $_POST['sectionID'];
    mysqli_query($connect, "INSERT INTO `Room`(`Title`,`SectionID`) VALUES ('$title',$SectionID)");
    $q = mysqli_query($connect, "SELECT ID,Title FROM Room WHERE SectionID=$SectionID");
    while ($r = mysqli_fetch_assoc($q)) {
        $Rooms[] = array('ID' => $r['ID'], 'Title' => $r['Title'], 'Count' => 0);
    }
    for ($i = 0; $i < count($Rooms); $i++) {
        $id = $Rooms[$i]['ID'];
        if ($result = mysqli_query($connect, "SELECT ID FROM Students WHERE RoomID=$id")) {
            $row_cnt = mysqli_num_rows($result);
            $Rooms[$i]['Count'] = $row_cnt;
        }
    }
    echo mysqli_error($connect);
    echo json_encode($Rooms);
}

/*****Students*****/
/*****/

```

```

if (isset($_POST['getStudentsByRoom'])) {
    $id = $_POST['roomId'];
    $Students = array();
    $q = mysqli_query($connect, "SELECT ID,Name,Image,University,Speciality,rating FROM Students WHERE RoomID=$
id");
    while ($r = mysqli_fetch_assoc($q)) {
        $Students[] = $r;
    }
    echo mysqli_error($connect);
    echo json_encode($Students);
}
if (isset($_POST['get_student_without_room'])) {
    $Students = array();
    $q = mysqli_query($connect, "SELECT ID,Name,Image,University,Speciality,rating FROM Students WHERE RoomID=0
");
    while ($r = mysqli_fetch_assoc($q)) {
        $Students[] = $r;
    }
    echo mysqli_error($connect);
    echo json_encode($Students);
}
if (isset($_POST['addUsersInRoom'])) {
    $roomId = $_POST['roomId'];
    $studentsIDs = $_POST['students'];
    for ($i = 0; $i < count($studentsIDs); $i++) {
        $id = $studentsIDs[$i];
        mysqli_query($connect, "UPDATE Students SET RoomID=$roomId WHERE ID=$id");
    }

    $Students = array();
    $q = mysqli_query($connect, "SELECT ID,Name,Image,University,Speciality,rating FROM Students WHERE RoomID=$
roomId");
    while ($r = mysqli_fetch_assoc($q)) {
        $Students[] = $r;
    }
    echo mysqli_error($connect);
    echo json_encode($Students);
}
if (isset($_POST['DeleteStudentFromRoom'])) {
    $ID = $_POST['id'];
    $roomId = $_POST['roomId'];
    mysqli_query($connect, "UPDATE Students SET RoomID=0 WHERE ID=$ID");
    $Students = array();
    $q = mysqli_query($connect, "SELECT ID,Name,Image,University,Speciality,rating FROM Students WHERE RoomID=$
roomId");
    while ($r = mysqli_fetch_assoc($q)) {
        $Students[] = $r;
    }
    echo mysqli_error($connect);
    echo json_encode($Students);
}
if (isset($_POST['DeleteStudent'])) {
    $ID = $_POST['id'];
    mysqli_query($connect, "DELETE FROM Students WHERE ID=$ID");
    echo mysqli_error($connect);
    $Students = array();
    $q = mysqli_query($connect, "SELECT ID,Name,Image,University,Speciality,rating FROM Students WHERE RoomID=0
");
    while ($r = mysqli_fetch_assoc($q)) {

```

```

    $Students[] = $r;
}
echo mysqli_error($connect);
echo json_encode($Students);
}
if (isset($_POST['create_new_student'])) {
    $uploaddir = './students'; // . - текущая папка где находится submit.php
    $userName = $_POST['studentName'];
    $studentUniv = $_POST['studentUniv'];
    $studentSpec = $_POST['studentSpec'];
    $studentRating = $_POST['studentRating'];
    $studentLogin = $_POST['studentLogin'];
    $studentPass = $_POST['studentPass'];
    // создадим папку если её нет
    if (!is_dir($uploaddir)) mkdir($uploaddir, 0777);

    $files = $_FILES; // полученные файлы
    $done_files = array();

    // переместим файлы из временной директории в указанную
    foreach ($files as $file) {
        $file_name = $file['name'];

        if (move_uploaded_file($file['tmp_name'], "$uploaddir/$file_name")) {
            $dir = str_replace('.', '', $uploaddir);
            $done_files = "$dir/$file_name";
        }
    }
    var_dump($done_files);

    $q = mysqli_query($connect, "SELECT ID FROM Students WHERE login='$studentLogin'");
    $row_cnt = mysqli_num_rows($q);
    if ($row_cnt == 0) {
        mysqli_query($connect, "INSERT INTO
        `Students`(`Name`, `RoomID`, `SectionID`, `HostelID`, `Image`, `University`, `Speciality`, `rating`, `login`, `pass`)
        VALUES ('$userName',0,0,0,'$done_files','$studentUniv','$studentSpec','$studentRating','$studentLogin','$studentPa
        ss')");
        echo mysqli_error($connect);
        $data = 1;
    } else {
        $data = 0;
    }
}

die(json_encode($data));
}

if (isset($_POST['searchRoom'])) {
    $search = $_POST['search'];
    $Rooms = array();
    $referal = trim(strip_tags(stripslashes(htmlspecialchars($search))));
    $q = mysqli_query($connect, "SELECT ID,Title,SectionID FROM Room WHERE Title LIKE '%$referal%'");
    while ($r = mysqli_fetch_assoc($q)) {
        $Rooms[] = $r;
    }
    $result = array();
    for ($i = 0; $i < count($Rooms); $i++) {
        $Section;
        $Hostel;
        $Users = array();
    }
}

```

```

$sectionID = $Rooms[$i]['SectionID'];
$q = mysqli_query($connect, "SELECT ID,Title,HostelID FROM Section WHERE ID=$sectionID");
$row_cnt = mysqli_num_rows($q);
if ($row_cnt > 0) {
    while ($r = mysqli_fetch_assoc($q)) {
        $Section = $r;
    }
    $HostelID = $Section['HostelID'];
    $q = mysqli_query($connect, "SELECT ID,Title,Address FROM hostels WHERE ID=$HostelID");
    while ($r = mysqli_fetch_assoc($q)) {
        $Hostel = $r;
    }
} else {
    $Section = $Hostel = 0;
}
$roomID = $Rooms[$i]['ID'];
$q = mysqli_query($connect, "SELECT ID,Name,Image,University,Speciality,rating FROM Students WHERE RoomID
=$roomID");
$row_cnt = mysqli_num_rows($q);
if ($row_cnt > 0) {
    while ($r = mysqli_fetch_assoc($q)) {
        $Users[] = $r;
    }
} else {
    $Users = 0;
}
$result[] = array('sections' => $Section, 'hostel' => $Hostel, 'users' => $Users, 'room' => $Rooms[$i]);
}
echo json_encode($result);
}
if (isset($_POST['searchUser'])) {
    $search = $_POST['search'];
    $Students = array();
    $referral = trim(strip_tags(strip_slashes(htmlspecialchars($search))));
    $q = mysqli_query($connect, "SELECT ID,Name,Image,RoomID,University,Speciality,rating FROM Students WHERE N
ame LIKE '%$referral%'");
    while ($r = mysqli_fetch_assoc($q)) {
        $Students[] = $r;
    }
    $result = array();
    for ($i = 0; $i < count($Students); $i++) {
        $room = $section = $hostel = "";
        $roomID = $Students[$i]['RoomID'];
        $q = mysqli_query($connect, "SELECT ID,Title,SectionID FROM Room WHERE ID=$roomID");
        $row_cnt = mysqli_num_rows($q);
        if ($row_cnt > 0) {
            while ($r = mysqli_fetch_assoc($q)) {
                $room = $r;
            }
            $sectionID = $room['SectionID'];
            $q = mysqli_query($connect, "SELECT ID,Title,HostelID FROM Section WHERE ID=$sectionID");
            $row_cnt = mysqli_num_rows($q);
            if ($row_cnt > 0) {
                while ($r = mysqli_fetch_assoc($q)) {
                    $section = $r;
                }
                $hostelID = $section['HostelID'];
                $q = mysqli_query($connect, "SELECT ID,Title,Address FROM hostels WHERE ID=$hostelID");
                $row_cnt = mysqli_num_rows($q);
            }
        }
    }
}

```

```

        if ($row_cnt > 0) {
            while ($r = mysqli_fetch_assoc($q)) {
                $hostel = $r;
            }
        } else {
            $hostel = 0;
        }
    } else {
        $section = $hostel = 0;
    }
} else {
    $room = $section = $hostel = 0;
}
$result[] = array('Student' => $Students[$i], 'Room' => $room, 'Section' => $section, 'Hostel' => $hostel);
}
echo json_encode($result);
}
if (isset($_POST['getStudentInfo'])) {
    if (isset($_SESSION['User'])) {
        $user = $_SESSION['User'];
        if ($user['type'] == 'student') {
            $id = $_SESSION['User']['ID'];
            $q = mysqli_query($connect, "SELECT ID,Name,Image,RoomID,University,Speciality,rating,login,pass FROM Students WHERE ID=$id");
            while ($r = mysqli_fetch_assoc($q)) {
                $Student = $r;
            }
            $result = array();
            $room = $section = $hostel = "";
            $roomID = $Student['RoomID'];
            $q = mysqli_query($connect, "SELECT ID,Title,SectionID FROM Room WHERE ID=$roomID");
            $row_cnt = mysqli_num_rows($q);
            if ($row_cnt > 0) {
                while ($r = mysqli_fetch_assoc($q)) {
                    $room = $r;
                }
                $sectionID = $room['SectionID'];
                $q = mysqli_query($connect, "SELECT ID,Title,HostelID FROM Section WHERE ID=$sectionID");
                $row_cnt = mysqli_num_rows($q);
                if ($row_cnt > 0) {
                    while ($r = mysqli_fetch_assoc($q)) {
                        $section = $r;
                    }
                    $hostelID = $section['HostelID'];
                    $q = mysqli_query($connect, "SELECT ID,Title,Address FROM hostels WHERE ID=$hostelID");
                    $row_cnt = mysqli_num_rows($q);
                    if ($row_cnt > 0) {
                        while ($r = mysqli_fetch_assoc($q)) {
                            $hostel = $r;
                        }
                    } else {
                        $hostel = 0;
                    }
                } else {
                    $section = $hostel = 0;
                }
            } else {
                $room = $section = $hostel = 0;
            }
        }
    }
}

```

```

    $result = array('Student' => $Student, 'Room' => $room, 'Section' => $section, 'Hostel' => $hostel);
    echo json_encode($result);
} else if ($user['type'] == 'watchman') {
    echo 0;
} else {
    echo 1;
}
} else {
    echo -1;
}
}
}

```

- admin.js

```

/*****Hostel*****/
/*****/
$(window).on('load', function () {
    $.ajax({
        url: '/assets/php/server.php',
        type: 'POST',
        data: { getHostels: "1" },
    })
    .done(function (data) {
        showHostels(JSON.parse(data));
    })
})
function showHostels(hostels) {
    console.log(hostels);
    $('#hostels>.db_res_list').empty();
    for (let i = 0; i < hostels.length; i++) {
        $('#hostels>.db_res_list').append(`
            <div class="db_res_item hostel_item" hostel-id="${hostels[i].ID}">
                
                
                <span class="title"> ${hostels[i].Title}</span>
                <span class="address">${hostels[i].Address}</span>
            </div>
        `);
    }
    $('#hostels>.db_res_list').append(`
        <div class="db_res_item hostel_item hostel_item_new" hostel-id="new">
            <span class="title"> + </span>
            <span class="address">Створити новий гуртожиток</span>
        </div>
    `);
}
$('#.db_res_list').on('click', '.hostel_item_new', function () {
    $('#create_new_hostel').fadeOut(200);
})
jQuery(function ($) {
    $(document).mouseup(function (e) { // событие клика по веб-документу
        var div = $(".create_new>div"); // тут указываем ID элемента
        if (!div.is(e.target) // если клик был не по нашему блоку
            && div.has(e.target).length === 0) { // и не по его дочерним элементам
            div.parent().fadeOut(200); // скрываем его
        }
    });
}

```



```

    }
  });
});
$('.create_hostel').on('click', function () {
  if ($('#new_hostel_title').val() != "") {
    if ($('#new_hostel_address').val() != "") {
      $.ajax({
        url: '/assets/php/server.php',
        type: 'POST',
        data: { CreateNewHostel: "1", title: $('#new_hostel_title').val(), address: $('#new_hostel_address').val() },
      })
      .done(function (data) {
        showHostels(JSON.parse(data));
        $('#create_new_hostel').fadeOut(200);
      })
    }
  }
})
$('.db_res_list').on('click', '.hostel_item>.delete_hostel', function () {
  let id = $(this).parent().attr('hostel-id');
  let title = $(this).parent().find('.title').text();
  let address = $(this).parent().find('.address').text();
  $('#hostels').append(`
    <div class="create_new delete_hostel" hostel-id="${id}">
    <div class="create_new_body">
      <h2>Ви впевнені що хочете видалити гуртожиток:</h2>
      <h3>${title}</h3>
      <h3>${address}</h3>
      <div class="btn_wrap">
        <div class="delete_hostel_btn delete_hostel_yes">Так</div>
        <div class="delete_hostel_btn delete_hostel_no">Hi</div>
      </div>
    </div>
  `);
  $('#hostels').find('.delete_hostel').fadeIn(200);
})
$('#hostels ').on('click', '.delete_hostel .delete_hostel_btn', function () {
  console.log($(this));
  if ($(this).hasClass('delete_hostel_no')) {
    $(this).parent().parent().parent().fadeOut(200);
  }
  else {
    let id = $(this).parent().parent().parent().attr('hostel-id');
    $.ajax({
      url: '/assets/php/server.php',
      type: 'POST',
      data: { DeleteHostel: "1", id: id },
    })
    .done(function (data) {
      showHostels(JSON.parse(data));
    })
  }
  $(this).parent().parent().parent().remove();
})
$('.db_res_list').on('click', '.hostel_item>.edit_hostel', function () {
  let id = $(this).parent().attr('hostel-id');
  let title = $(this).parent().find('.title').text();

```

```

let address = $(this).parent().find('.address').text();
$('.hostels').append(`
<div class="create_new edit_hostel" hostel-id="${id}">
<div class="create_new_body">
  <h2>Редагування гуртожитка:</h2>
  <div class="input_wrap">
    <label>Вкажіть назву гуртожитка</label>
    <input type="text" class="edit_hostel_title" value="${title}"/>
  </div>
  <div class="input_wrap">
    <label>Вкажіть адресу гуртожитка:</label>
    <input type="text" class="edit_hostel_address" value="${address}"/>
  </div>
  <div class="update_hostel create_new_btn">Оновити гуртожиток</div>
</div>
</div>
`);
$('.hostels').find('.edit_hostel').fadeIn(200);
})
$('.hostels').on('click', '.edit_hostel .update_hostel', function () {
  let id = $(this).parent().parent().attr('hostel-id');
  if ($(this).parent().find('.edit_hostel_title').val() != '') {
    if ($(this).parent().find('.edit_hostel_address').val() != '') {
      $.ajax({
        url: '/assets/php/server.php',
        type: 'POST',
        data: { UpdateHostel: "1", title: $(this).parent().find('.edit_hostel_title').val(), address: $(this).parent().find('.edit_hostel_address').val(), id: id },
      })
      .done(function (data) {
        showHostels(JSON.parse(data));
      })
      $(this).parent().parent().remove();
    }
  }
})
$('.db_res_list').on('click', '.hostel_item', function (e) {
  if ($(this).is(e.target)) {
    let id = $(this).attr('hostel-id');
    let title = $(this).find('.title').text();
    let address = $(this).find('.address').text();
    if (id != 'new') {
      $('.hostels').fadeOut(200);
      getSectionsByHostel(id);
      $('.sections').attr('hostel-id', id);
      $('.sections>h2').text(`Оберіть секцію з : ${title}, за адресою: ${address}`).attr({ 'hostel': title, 'address': address });
      $('.go_back').removeClass('now_hostel').addClass('now_section').fadeIn(200);
    }
  }
})
/*****Sections*****/
/*****/

function getSectionsByHostel(id) {
  $.ajax({
    url: '/assets/php/server.php',
    type: 'POST',

```

```

    data: { getSectionsByHostel: "1", id: id },
  })
  .done(function (data) {
    console.log(data);
    showSections(JSON.parse(data));
    console.log(id);
    console.log(JSON.parse(data));
  })
}
function showSections(sections) {
  console.log(sections);
  $('<div>.sections>.db_res_list').empty();
  for (let i = 0; i < sections.length; i++) {
    $('<div>.sections>.db_res_list').append(`
      <div class="db_res_item section_item" section-id="${sections[i].ID}">
        
        
        <span class="title"> ${sections[i].Title}</span>
        <span class="address">Кімнат - ${sections[i].Count}</span>
      </div>
    `);
  }
  $('<div>.sections>.db_res_list').append(`
    <div class="db_res_item section_item section_item_new" section-id="new">
      <span class="title"> + </span>
      <span class="address">Створити нову секцію</span>
    </div>
  `);
  setTimeout(() => {
    $('<div>.sections').fadeOut(200);
  }, 210)
}
$('<div>.go_back').on('click', function () {
  if ($(<div>this).hasClass('now_section')) {
    $(<div>this).removeClass('now_section').addClass('now_hostel').fadeOut(200);
    $('<div>.sections').fadeOut(200);
    setTimeout(() => {
      $('<div>.hostels').fadeIn(200);
    }, 210)
  }
  if ($(<div>this).hasClass('now_room')) {
    $(<div>this).removeClass('now_room now_section now_hostel').addClass('now_section');
    $('<div>.rooms').fadeOut(200);
    setTimeout(() => {
      $('<div>.sections').fadeIn(200);
    }, 210)
  }
  if ($(<div>this).hasClass('now_students')) {
    $(<div>this).removeClass('now_room now_section now_hostel now_students').addClass('now_room');
    $('<div>.students').fadeOut(200);
    setTimeout(() => {
      $('<div>.rooms').fadeIn(200);
    }, 210)
  }
})
}
})

```

```

$('.db_res_list').on('click', '.section_item>.delete_section', function () {
  let id = $(this).parent().attr('section-id');
  let title = $(this).parent().find('.title').text();
  let address = $(this).parent().find('.address').text();
  $('.sections').append(`
    <div class="create_new delete_section" section-id="${id}">
    <div class="create_new_body">
      <h2>Ви впевнені що хочете видалити секцію?</h2>
      <h3>${title}</h3>
      <h3>${address}</h3>
      <div class="btn_wrap">
        <div class="delete_section_btn delete_section_yes">Так</div>
        <div class="delete_section_btn delete_section_no">Ні</div>
      </div>
    </div>
  `);
  $('.sections').find('.delete_section').fadeOut(200);
})
$('.sections ').on('click', '.delete_section .delete_section_btn', function () {
  console.log($(this));
  if ($(this).hasClass('delete_section_no')) {
    $(this).parent().parent().parent().fadeOut(200);
  }
  else {
    let id = $(this).parent().parent().parent().attr('section-id');
    $.ajax({
      url: '/assets/php/server.php',
      type: 'POST',
      data: { DeleteSection: "1", id: id },
    })
    .done(function (data) {
      console.log(data);
      showSections(JSON.parse(data));
    })
  }
  $(this).parent().parent().parent().remove();
})
$('.db_res_list').on('click', '.section_item .edit_section', function () {
  console.log(123123123);
  let id = $(this).parent().attr('section-id');
  let title = $(this).parent().find('.title').text();
  $('.sections').append(`
    <div class="create_new edit_section" section-id="${id}">
    <div class="create_new_body">
      <h2>Редагування секції:</h2>
      <div class="input_wrap">
        <label>Вкажіть назву Секції</label>
        <input type="text" class="edit_section_title" value="${title}"/>
      </div>
      <div class="update_section create_new_btn">Оновити секцію</div>
    </div>
  `);
  $('.sections').find('.edit_section').fadeOut(200);
})
$('.sections').on('click', '.edit_section .update_section', function () {
  let id = $(this).parent().parent().attr('section-id');
  if ($(this).parent().find('.edit_section_title').val() != '') {

```



```

function getRoomsBySection(sectionId, hostelId) {
  $.ajax({
    url: '/assets/php/server.php',
    type: 'POST',
    data: { getRoomsBySection: "1", sectionId: sectionId },
  })
  .done(function (data) {
    showRooms(JSON.parse(data));
  })
}

function showRooms(rooms) {
  console.log(rooms);
  $('<rooms>.db_res_list').empty();
  for (let i = 0; i < rooms.length; i++) {
    $('<rooms>.db_res_list').append(`
      <div class="db_res_item room_item" room-id="${rooms[i].ID}">
      
      
      <span class="title"> ${rooms[i].Title}</span>
      <span class="address"> мешканці - ${rooms[i].Count}</span>
    </div>
  `);
  }
  $('<rooms>.db_res_list').append(`
    <div class="db_res_item room_item room_item_new" room-id="new">
      <span class="title"> + </span>
      <span class="address"> Створити нову Кімнату</span>
    </div>
  `);
  setTimeout(() => {
    $('<rooms>').fadeOut(200);
  }, 210)
}

$('<db_res_list>').on('click', '<room_item>.delete_room', function () {
  let id = $(this).parent().attr('room-id');
  let title = $(this).parent().find('.title').text();
  let address = $(this).parent().find('.address').text();
  $('<rooms>').append(`
    <div class="create_new delete_room" room-id="${id}">
    <div class="create_new_body">
      <h2>Ви впевнені що хочете видалити кімнату?</h2>
      <h3>${title}</h3>
      <h3>${address}</h3>
      <div class="btn_wrap">
        <div class="delete_room_btn delete_room_yes">Так</div>
        <div class="delete_room_btn delete_room_no">Hi</div>
      </div>
    </div>
  `);
  $('<rooms>').find('<delete_room>').fadeOut(200);
})

$('<rooms>').on('click', '<delete_room>.delete_room_btn', function () {
  console.log($(this));
  if ($(this).hasClass('delete_room_no')) {
    $(this).parent().parent().parent().fadeOut(200);
  }
  else {

```

```

let id = $(this).parent().parent().parent().attr('room-id');
$.ajax({
  url: '/assets/php/server.php',
  type: 'POST',
  data: { DeleteRoom: "1", id: id },
})
.done(function (data) {
  console.log(data);
  showRooms(JSON.parse(data));
})
}
$(this).parent().parent().parent().remove();
})
$('.db_res_list').on('click', '.room_item .edit_room', function () {
  let id = $(this).parent().attr('room-id');
  let title = $(this).parent().find('.title').text();
  $('.rooms').append(`
<div class="create_new edit_room" room-id="${id}">
<div class="create_new_body">
  <h2>Редагування Кімнати:</h2>
  <div class="input_wrap">
    <label>Вкажіть назву Кімнати</label>
    <input type="text" class="edit_room_title" value="${title}"/>
  </div>
  <div class="update_room create_new_btn">Оновити Кімнату</div>
</div>
</div>
`);
  $('.rooms').find('.edit_room').fadeIn(200);
})
$('.rooms').on('click', '.edit_room .update_room', function () {
  let id = $(this).parent().parent().attr('room-id');
  if ($(this).parent().find('.edit_room_title').val() != "") {

    $.ajax({
      url: '/assets/php/server.php',
      type: 'POST',
      data: { UpdateRoom: "1", title: $(this).parent().find('.edit_room_title').val(), id: id },
    })
    .done(function (data) {
      showRooms(JSON.parse(data));
    })
    $(this).parent().parent().remove();
  }
})
$('.db_res_list').on('click', '.room_item_new', function () {
  $('.create_new_room').fadeIn(200);
})
$('.create_room').on('click', function () {
  let sectionID = $(this).parent().parent().parent().attr('section-id');
  if ($('#new_room_title').val() != "") {
    $.ajax({
      url: '/assets/php/server.php',
      type: 'POST',
      data: { CreateNewRoom: "1", title: $('#new_room_title').val(), sectionID: sectionID },
    })
    .done(function (data) {
      showRooms(JSON.parse(data));
      $('#create_new_room').fadeOut(200);
    });
  }
});

```

```

    })
  }
})
$('.db_res_list').on('click', '.room_item', function (e) {
  if ($(this).is(e.target)) {
    let roomId = $(this).attr('room-id');
    let roomTitle = $(this).find('.title').text();
    let hostelId = $(this).parent().parent().find('h2').attr('hostelID');
    let sectionId = $(this).parent().parent().find('h2').attr('sectionID');
    let sectionTitle = $(this).parent().parent().find('h2').attr('section');
    let hostelTitle = $(this).parent().parent().find('h2').attr('hostel');
    let hostelAddress = $(this).parent().parent().find('h2').attr('address');
    if (roomId != 'new') {
      $('.rooms').fadeOut(200);
      if ($(this).parent().parent().parent().hasClass('edit_db')) {
        getStudentsByRoom(roomId);
      }
      else {
        getStudentsWithEditByRoom(roomId);
      }

      $('.students').attr({ 'hostel-id': hostelId, 'section-id': sectionId, 'room-id': roomId });
      $('.students>h2').text(`Перелік мешканців : ${roomTitle}, ${sectionTitle}, ${hostelTitle} за адресою: ${hostelAddress}`);
      $('.students').attr(
        {
          'section': sectionTitle,
          'hostel': hostelTitle,
          'address': hostelAddress,
          'roomTitle': roomTitle
        }
      );
      $('.go_back').removeClass('now_hostel now_section now_room').addClass('now_students').fadeIn(200);
    }
  }
})
/*****Students*****/
/*****/

function getStudentsByRoom(roomId) {
  $.ajax({
    url: '/assets/php/server.php',
    type: 'POST',
    data: { getStudentsByRoom: "1", roomId: roomId },
  })
  .done(function (data) {
    console.log(data);
    showStudents(JSON.parse(data));
  })
}

function getStudentsWithEditByRoom(roomId) {
  $.ajax({
    url: '/assets/php/server.php',
    type: 'POST',
    data: { getStudentsByRoom: "1", roomId: roomId },
  })
  .done(function (data) {
    console.log(data);
    showStudentsWithEdit(JSON.parse(data));
  })
}

```



```

    })
  }
function showStudents(students) {
  console.log(students);
  $('>.students>.users_list').empty();
  for (let i = 0; i < students.length; i++) {
    let raingClass = 'green';
    if (students[i].rating < 50) {
      raingClass = "red";
    }
    let ImagePath = "";
    if (students[i].Image == "") {
      ImagePath = '/assets/img/user.png';
    }
    else {
      ImagePath = '/assets/php' + students[i].Image;
    }
    $('>.students>.users_list').append(`
    <div class="db_res_list_item student_item" student-id="${students[i].ID}">
      
      <span class="student_name"> ${students[i].Name}</span>
      <span class="student_un"> ${students[i].University}</span>
      <span class="student_spec"> ${students[i].Speciality}</span>
      <span class="student_rating ${raingClass}">${students[i].rating}</span>
    </div>
    `);
  }
  setTimeout(() => {
    $('>.students').fadeOut(200);
  }, 210)
}
function showStudentsWithEdit(students) {
  console.log(students);
  $('>.students>.users_list').empty();
  for (let i = 0; i < students.length; i++) {
    let raingClass = 'green';
    if (students[i].rating < 50) {
      raingClass = "red";
    }
    let ImagePath = "";
    if (students[i].Image == "") {
      ImagePath = '/assets/img/user.png';
    }
    else {
      ImagePath = '/assets/php' + students[i].Image;
    }
    $('>.students>.users_list').append(`
    <div class="db_res_list_item student_item" student-id="${students[i].ID}">
      
      <span class="student_name"> ${students[i].Name}</span>
      <span class="student_un"> ${students[i].University}</span>
      <span class="student_spec"> ${students[i].Speciality}</span>
      <span class="student_rating ${raingClass}">${students[i].rating}</span>
      
      
    </div>
    `);
  }
  $('>.students>.users_list').append(`

```

```

<div class="db_res_list_item student_item student_item_new_in_room" student-id="new">
  
</div>
`);
setTimeout(() => {
  $('.students').fadeOut(200);
}, 210)
}

$('.students').on('click', '.student_item_new_in_room', function () {
  console.log(333112123123213);
  let id = $(this).parent().parent().attr('room-id');
  let title = $(this).parent().find('.title').text();
  $('.students').append(`
  <div class="create_new add_student_in_room" room-id="${id}">
  <div class="create_new_body">
    <h2>Заселити нового студента в кімнату:</h2>
    <div class="input_wrap">
      <label>Оберіть студента без кімнати з переліку:</label>
      <div class="students_without_room_list">
        </div>
      </div>
      <div class="add_new_student_in_room_btn create_new_btn">Оновити Кімнату</div>
    </div>
  </div>
  `);
  get_student_without_room(students_without_room_list_for_add);
  $('.students').find('.add_student_in_room').fadeOut(200);
})

function get_student_without_room(func) {
  $.ajax({
    url: '/assets/php/server.php',
    type: 'POST',
    data: { get_student_without_room: "1", },
  })
  .done(function (data) {

    func(JSON.parse(data));
  })
}

function students_without_room_list_for_add(students) {
  console.log(students);
  $('.students>.add_student_in_room>.create_new_body>.input_wrap>.students_without_room_list').empty();
  for (let i = 0; i < students.length; i++) {
    let raingClass = 'green';
    if (students[i].rating < 50) {
      raingClass = "red";
    }
    let imagePath = "";
    console.log(students[i].Image == "");
    if (students[i].Image == "") {
      imagePath = '/assets/img/user.png';
    }
    else {
      imagePath = '/assets/php' + students[i].Image;
    }
  }
  $('.students>.add_student_in_room>.create_new_body>.input_wrap>.students_without_room_list').append(`

```

```

<div class="new_student_in_room_item" student-id="{students[i].ID}">
  
  <span class="student_name"> {students[i].Name}</span>
  <span class="student_un"> {students[i].University}</span>
  <span class="student_spec"> {students[i].Speciality}</span>
  <span class="student_rating {raingClass}">{students[i].rating}</span>
  <div class="students_in_room_check"></div>
</div>
`);
}
}
$('.students').on('click', '.add_student_in_room .new_student_in_room_item .students_in_room_check', function () {
  $(this).toggleClass('checked');
  $(this).parent().toggleClass('checked');
})
$('.students').on('click', '.add_student_in_room .add_new_student_in_room_btn ', function () {
  let students = $(this).parent().find('.input_wrap').find('.students_without_room_list').find('.new_student_in_room_i
tem.checked');
  let roomID = $(this).parent().parent().attr('room-id');
  let selectedIDS = [];
  for (let i = 0; i < students.length; i++) {
    selectedIDS[i] = Number(students[i].attributes[1].nodeValue);
  }
  console.log(selectedIDS);
  $.ajax({
    url: '/assets/php/server.php',
    type: 'POST',
    data: { addUsersInRoom: "1", roomID: roomID, students: selectedIDS },
  })
  .done(function (data) {
    showStudentsWithEdit(JSON.parse(data));
    $('.students').find('.add_student_in_room').remove();
  })
})
$('.users_list').on('click', '.student_item .delete_student_from_room', function () {
  let studentID = $(this).parent().attr('student-id');
  let userName = $(this).parent().find('.student_name').text();
  let roomTitle = $(this).parent().parent().parent().find('h2').attr('roomtitle');
  $('.students').append(`
<div class="create_new_delete_user_from_room" student-id="{studentID}">
<div class="create_new_body">
  <h2>Ви впевнені що хочете виселити {userName}</h2>
  <h2>з {roomTitle}</h2>
  <div class="btn_wrap">
    <div class="delete_user_from_room_btn delete_user_from_room_btn_yes">Так</div>
    <div class="delete_user_from_room_btn delete_user_from_room_btn_no">Hi</div>
  </div>
</div>
`);
  $('.students').find('.delete_user_from_room').fadeOut(200);
})
$('.students').on('click', '.delete_user_from_room .delete_user_from_room_btn', function () {
  console.log($(this));
  if ($(this).hasClass('delete_user_from_room_btn_no')) {
    $(this).parent().parent().parent().fadeOut(200);
  }
  else {

```

```

let id = $(this).parent().parent().parent().attr('student-id');

let roomID = $(this).parent().parent().parent().parent().attr('room-id');
console.log(id, roomID);
$.ajax({
  url: '/assets/php/server.php',
  type: 'POST',
  data: { DeleteStudentFromRoom: "1", id: id, roomID: roomID },
})
  .done(function (data) {
    console.log(data);
    showStudentsWithEdit(JSON.parse(data));
  })
}
$(this).parent().parent().parent().remove();
})

```

```

function clear_hierarchy() {
  $('.sections').fadeOut(200);
  $('.rooms').fadeOut(200);
  $('.students').fadeOut(200);
  $('.go_back').removeClass('now_section now_room now_students').addClass('now_hostel').fadeOut(200);
  setTimeout(() => {
    $('.hostels').fadeIn(200);
  }, 210)
}

$('.menu_item').on('click', function () {
  clear_hierarchy();
  if ($(this).attr('menu-type') == 'cr') {
    $('.edit_db').fadeOut(200);
    setTimeout(function () {
      $('.change_room').fadeIn(200);
    }, 210)
  }
}

```

```

}
if ($(this).attr('menu-type') == 'db') {
    $('.change_room').fadeOut(200);
    setTimeout(function () {
        $('.edit_db').fadeIn(200);
    }, 210)
}
$('.menu_item').removeClass('menu_item_active');
$(this).addClass('menu_item_active');
})

$('.select_event>div').on('click', function () {
    $('.select_event>div').removeClass('selected_event');
    $(this).addClass('selected_event');
    if ($(this).hasClass('event_change_room')) {
        $('.select_event_body>div').fadeOut(200);
        $('.change_room_block').fadeIn(100);
    }
    if ($(this).hasClass('event_list_users_without_room')) {
        $('.select_event_body>div').fadeOut(200);
        $('.list_users_without_room_body').fadeIn(100);
        usersWithoutRoomInit();
    }
    if ($(this).hasClass('event_create_user')) {
        $('.select_event_body>div').fadeOut(200);
        $('.create_user_body').fadeIn(100);
    }
})
function usersWithoutRoomInit() {
    get_student_without_room(setUsersInList);
}
function setUsersInList(students) {
    console.log(students);
    $('.list_users_without_room_body .list_users_without_room_body_list').empty();
    for (let i = 0; i < students.length; i++) {
        let raingClass = 'green';
        if (students[i].rating < 50) {
            raingClass = "red";
        }
        let ImagePath = "";
        if (students[i].Image == "") {
            ImagePath = '/assets/img/user.png';
        }
        else {
            ImagePath = '/assets/php' + students[i].Image;
        }
        $('.list_users_without_room_body .list_users_without_room_body_list').append(`
<div class="db_res_list_item student_item" student-id="${students[i].ID}">
    
    <span class="student_name"> ${students[i].Name}</span>
    <span class="student_un"> ${students[i].University}</span>
    <span class="student_spec"> ${students[i].Speciality}</span>
    <span class="student_rating ${raingClass}">${students[i].rating}</span>
    

```

```

    </div>
    `);
}
}
$('.list_users_without_room_body').on('click', '.student_item .delete_student', function () {
    let studentID = $(this).parent().attr('student-id');
    let userName = $(this).parent().find('.student_name').text();
    $('.list_users_without_room_body').append(`
    <div class="create_new delete_user" student-id="${studentID}">
    <div class="create_new_body">
        <h2>Ви впевнені що хочете видалити ${userName}</h2>
        <div class="btn_wrap">
            <div class="delete_user_btn delete_user_btn_yes">Так</div>
            <div class="delete_user_btn delete_user_btn_no">Hi</div>
        </div>
    </div>
    </div>
    `);
    $('.list_users_without_room_body').find('.delete_user').fadeIn(200);
})
$('.list_users_without_room_body').on('click', '.delete_user .delete_user_btn', function () {
    console.log($(this));
    if ($(this).hasClass('delete_user_btn_no')) {
        $(this).parent().parent().parent().fadeOut(200);
    }
    else {
        let id = $(this).parent().parent().parent().attr('student-id');
        console.log(id);
        $.ajax({
            url: '/assets/php/server.php',
            type: 'POST',
            data: { DeleteStudent: "1", id: id },
        })
        .done(function (data) {
            console.log(data);
            get_student_without_room(setUsersInList);
        })
    }
    $(this).parent().parent().parent().remove();
})
$(document).ready(function () {
    $.uploadPreview({
        input_field: ".new_user_image", // По умолчанию: .image-upload
        preview_box: ".image_preview", // По умолчанию: .image-preview
        label_field: ".new_user_image_label", // По умолчанию: .image-label
        label_default: "Зображення", // По умолчанию: Choose File
        label_selected: "Змінити", // По умолчанию: Change File
        no_label: false, // По умолчанию: false
        success_callback: null // По умолчанию: null
    });
});
var files; // переменная. будет содержать данные файлов

// заполняем переменную данными, при изменении значения поля file
$('input[type=file]').on('change', function () {
    files = this.files;
});
$('.create_student').on('click', function (event) {

```

```

event.stopPropagation(); // остановка всех текущих JS событий
event.preventDefault(); // остановка дефолтного события для текущего элемента - клик для <a> тега
if (typeof files == 'undefined') return;
var data = new FormData();
$.each(files, function (key, value) {
    data.append(key, value);
});
data.append('studentName', $('#new_user_name').val());
data.append('studentUniv', $('#new_user_univ').val());
data.append('studentSpec', $('#new_user_spec').val());
data.append('studentRating', $('#new_user_rating').val());
data.append('studentLogin', $('#new_user_login').val());
data.append('studentPass', $('#new_user_pass').val());
data.append('create_new_student', 1);
$.ajax({
    url: '/assets/php/server.php',
    type: 'POST', // важно!
    data: data,
    cache: false,
    dataType: 'json',
    processData: false,
    contentType: false,
    success: function (data) {

    },
});
$('#new_user_name').val("");
$('#new_user_univ').val("");
$('#new_user_spec').val("");
$('#new_user_rating').val("");
$('#new_user_image').val("");

})

$('#.exit').on('click',function(){
$.ajax({
    url: '/assets/php/server.php',
    type: 'POST',
    data: { exit: "1" },
})
.done(function (data) {
    window.location.href='./';
})
})
})

```

- main.js

```

$.ajax({
    url: '/assets/php/server.php',
    type: 'POST',
    data: { getCurrUser: "1" },
})
.done(function (data) {
    if (data != 0) {
        let user = JSON.parse(data);
        //console.log(user.length);
        if (user.type == 'student') {

```

```

        window.location.href = "/student.html";
    }
    else if (user.type == "admin") {
        window.location.href = "/admin.html";
    }
    else if (user.type == "watchman") {
        window.location.href = "/watchman.html";
    }
}
})
$('.signin-form__submit-button').on('click', function (e) {
    e.preventDefault();
    if ($('#signin-form__field[name="login"]').val() != "" && $('#signin-form__field[name="password"]').val() != "") {
        let login = $('#signin-form__field[name="login"]').val();
        let pass = $('#signin-form__field[name="password"]').val();
        $.ajax({
            url: '/assets/php/server.php',
            type: 'POST',
            data: {auth: "1",login:login,pass:pass },
        })
        .done(function (data) {
            if (data == 2) {
                window.location.href = "/student.html";
            }
            else if (data == 1) {
                window.location.href = "/admin.html";
            }
            else if (data == 0) {
                window.location.href = "/watchman.html";
            }
            else{
                alert('користувача з такими даними не знайдено');
            }
        })
    }
})

```

student.js

```

$(window).on('load', function () {
    $.ajax({
        url: '/assets/php/server.php',
        type: 'POST',
        data: { getStudentInfo: "1" },
    })
    .done(function (data) {
        if(data===-1){
            window.location.href='../';
        }
        else if(data==0){
            window.location.href='../watchman.html';
        }
        else if(data==1){
            window.location.href='../admin.html';
        }
        else{
            setUserInfo(JSON.parse(data));
        }
    })
}

```



```

    };

    })
  })
function setUserInfo(arr) {
  let usersData = "";
  let item="";
  let raingClass = 'green';
  if (arr.Student.rating < 50) {
    raingClass = "red";
  }
  let ImagePath = "";
  if (arr.Student.Image == "") {
    ImagePath = '/assets/img/user.png';
  }
  else {
    ImagePath = '/assets/php' + arr.Student.Image;
  }
  let location = ``;
  usersData += `<div class="search_user_item"> <div class="user_item_left"></div>
<div class="user_item_right"><span class="student_name"> ${arr.Student.Name}</span>
  <span class="student_un"> ${arr.Student.University}</span>
  <span class="student_spec"> ${arr.Student.Speciality}</span>
  <span class="student_rating ${raingClass}">${arr.Student.rating}</span>
  <span class="student_login">${arr.Student.login}</span>
  <span class="student_location"> ${arr.Hostel == 0 & arr.Section == 0 & arr.Room == 0 ? ' Студент без кімнат
и' : arr.Room.Title + ', ' + arr.Section.Title + ', ' + arr.Hostel.Title + ' за адресою -
' + arr.Hostel.Address}</span></div></div>`;

  item += `
  <div class="search_result_item">
    <div class="users_list">
      ${usersData}
    </div>
  </div>`;
  $('student_page').append(item);
}
$('.exit').on('click',function(){
$.ajax({
  url: '/assets/php/server.php',
  type: 'POST',
  data: { exit: "1" },
})
.done(function (data) {
  window.location.href='./';
})
})
})

```

- watchman.js

```

/*****Hostel*****/
/*****/
$(window).on('load', function () {
$.ajax({
  url: '/assets/php/server.php',
  type: 'POST',

```

```

    data: { getHostels: "1" },
  })
  .done(function (data) {
    showHostels(JSON.parse(data));
  })
})
function showHostels(hostels) {
  console.log(hostels);
  $('.hostels>.db_res_list').empty();
  for (let i = 0; i < hostels.length; i++) {
    $('.hostels>.db_res_list').append(`
      <div class="db_res_item hostel_item" hostel-id="${hostels[i].ID}">
        <span class="title"> ${hostels[i].Title}</span>
        <span class="address">${hostels[i].Address}</span>
      </div>
    `);
  }
}
$('.db_res_list').on('click', '.hostel_item_new', function () {
  $('.create_new_hostel').fadeOut(200);
})
jQuery(function ($) {
  $(document).mouseup(function (e) { // событие клика по веб-документу
    var div = $(".create_new>div"); // тут указываем ID элемента
    if (!div.is(e.target) // если клик был не по нашему блоку
      && div.has(e.target).length === 0) { // и не по его дочерним элементам
      div.parent().fadeOut(200); // скрываем его
    }
  });
});
$('.db_res_list').on('click', '.hostel_item', function (e) {
  if ($(this).is(e.target)) {
    let id = $(this).attr('hostel-id');
    let title = $(this).find('.title').text();
    let address = $(this).find('.address').text();
    if (id !== 'new') {
      $('.hostels').fadeOut(200);
      getSectionsByHostel(id);
      $('.sections').attr('hostel-id', id);
      $('.sections>h2').text(`Оберіть секцію з : ${title}, за адресою: ${address}`).attr({ 'hostel': title, 'address': address });
    }
    $('.go_back').removeClass('now_hostel').addClass('now_section').fadeIn(200);
  }
}
})
/*****Sections*****/
/*****/

function getSectionsByHostel(id) {
  $.ajax({
    url: '/assets/php/server.php',
    type: 'POST',
    data: { getSectionsByHostel: "1", id: id },
  })
  .done(function (data) {
    console.log(data);
    showSections(JSON.parse(data));
  });
}

```

```

        console.log(id);
        console.log(JSON.parse(data));
    })
}
function showSections(sections) {
    console.log(sections);
    $('.sections>.db_res_list').empty();
    for (let i = 0; i < sections.length; i++) {
        $('.sections>.db_res_list').append(`
            <div class="db_res_item section_item" section-id="${sections[i].ID}">
                <span class="title"> ${sections[i].Title}</span>
                <span class="address">Кімнат - ${sections[i].Count}</span>
            </div>
        `);
    }
    setTimeout(() => {
        $('.sections').fadeOut(200);
    }, 210)
}
$('.go_back').on('click', function () {
    if ($(this).hasClass('now_section')) {
        $(this).removeClass('now_section').addClass('now_hostel').fadeOut(200);
        $('.sections').fadeOut(200);
        setTimeout(() => {
            $('.hostels').fadeIn(200);
        }, 210)
    }
    if ($(this).hasClass('now_room')) {
        $(this).removeClass('now_room now_section now_hostel').addClass('now_section');
        $('.rooms').fadeOut(200);
        setTimeout(() => {
            $('.sections').fadeIn(200);
        }, 210)
    }
    if ($(this).hasClass('now_students')) {
        $(this).removeClass('now_room now_section now_hostel now_students').addClass('now_room');
        $('.students').fadeOut(200);
        setTimeout(() => {
            $('.rooms').fadeIn(200);
        }, 210)
    }
})
$('.db_res_list').on('click', '.section_item', function (e) {
    if ($(this).is(e.target)) {
        let sectionId = $(this).attr('section-id');
        let hostelId = $(this).parent().parent().attr('hostel-id');
        let sectionTitle = $(this).find('.title').text();
        let hostelTitle = $(this).parent().parent().find('h2').attr('hostel');
        let hostelAddress = $(this).parent().parent().find('h2').attr('address');
        if (sectionId !== 'new') {
            $('.sections').fadeOut(200);
            getRoomsBySection(sectionId, hostelId);
            $('.rooms').attr({ 'hostel-id': hostelId, 'section-id': sectionId, });
            $('.rooms>h2').text('Оберіть кімнату : ${sectionTitle}, ${hostelTitle} за адресою: ${hostelAddress}').attr(
                {

```

```

        'section': sectionTitle,
        'hostel': hostelTitle,
        'address': hostelAddress,
        'sectionID': sectionId,
        'hostelID': hostelId,
    }
    );
    $('#go_back').removeClass('now_hostel now_section').addClass('now_room').fadeIn(200);
}
}
})

/*****ROOMS*****/
/*****/

function getRoomsBySection(sectionId, hostelId) {
    $.ajax({
        url: '/assets/php/server.php',
        type: 'POST',
        data: { getRoomsBySection: "1", sectionId: sectionId },
    })
    .done(function (data) {
        showRooms(JSON.parse(data));
    })
}

function showRooms(rooms) {
    console.log(rooms);
    $('#rooms>.db_res_list').empty();
    for (let i = 0; i < rooms.length; i++) {
        $('#rooms>.db_res_list').append(`
            <div class="db_res_item room_item" room-id="${rooms[i].ID}">
                <span class="title"> ${rooms[i].Title}</span>
                <span class="address"> мешканці - ${rooms[i].Count}</span>
            </div>
        `);
    }
    setTimeout(() => {
        $('#rooms').fadeIn(200);
    }, 210)
}

$('#db_res_list').on('click', '.room_item', function (e) {
    if ($(this).is(e.target)) {
        let roomId = $(this).attr('room-id');
        let roomTitle = $(this).find('.title').text();
        let hostelId = $(this).parent().parent().find('h2').attr('hostelID');
        let sectionId = $(this).parent().parent().find('h2').attr('sectionID');
        let sectionTitle = $(this).parent().parent().find('h2').attr('section');
        let hostelTitle = $(this).parent().parent().find('h2').attr('hostel');
        let hostelAddress = $(this).parent().parent().find('h2').attr('address');
        if (roomId != 'new') {
            $('#rooms').fadeOut(200);
            if ($(this).parent().parent().parent().hasClass('edit_db')) {
                getStudentsByRoom(roomId);
            }
            else {
                getStudentsWithEditByRoom(roomId);
            }
        }
    }
}

```



```

        <span class="student_spec"> ${students[i].Speciality}</span>
        <span class="student_rating ${raingClass}">${students[i].rating}</span>
    </div>
    `);
}
setTimeout(() => {
    $('.students').fadeOut(200);
}, 210)
}
function showStudentsWithEdit(students) {
    console.log(students);
    $('.students>.users_list').empty();
    for (let i = 0; i < students.length; i++) {
        let raingClass = 'green';
        if (students[i].rating < 50) {
            raingClass = "red";
        }
        let imagePath = "";
        if (students[i].Image == "") {
            imagePath = '/assets/img/user.png';
        }
        else {
            imagePath = '/assets/php' + students[i].Image;
        }
        $('.students>.users_list').append(`
        <div class="db_res_list_item student_item" student-id="${students[i].ID}">
            
            <span class="student_name"> ${students[i].Name}</span>
            <span class="student_un"> ${students[i].University}</span>
            <span class="student_spec"> ${students[i].Speciality}</span>
            <span class="student_rating ${raingClass}">${students[i].rating}</span>
        </div>
        `);
    }
    setTimeout(() => {
        $('.students').fadeOut(200);
    }, 210)
}
$('.search_btn').on('click', function () {
    if ($('#search_input').val() != "") {
        let search = $('#search_input').val();
        if ($('#search_input').val().search(/\d/) != -1) {

            $.ajax({
                url: '/assets/php/server.php',
                type: 'POST',
                data: { searchRoom: "1", search: $('#search_input').val() },
            })
            .done(function (data) {
                console.log(JSON.parse(data));
                setRoomsInSerchResult(JSON.parse(data), search);
            })
        }
        else {
            $.ajax({
                url: '/assets/php/server.php',
                type: 'POST',

```

```

        data: { searchUser: "1", search: $('#search_input').val() },
    })
    .done(function (data) {
        setUsersInSerchResult(JSON.parse(data), search);
        console.log(JSON.parse(data));
    })
}
setTimeout(() => {
    $('#search_result').fadeIn(200);
}, 210);
}

})
$('#close_search_result').on('click', function () {
    $('#search_result').fadeOut(200);
    $('#search_input').val("");
    setTimeout(() => {
        $('#edit_db').fadeIn(200);

    }, 210);
})
function setUsersInSerchResult(arr, search) {
    $('#search_result .search_result_body').empty();
    $('#search_result>h2').text('Результати пошуку за запитом - ' + search + ' :');
    $('#edit_db').fadeOut(200);
    let item = ``;
    let usersData = "";
    for (let i = 0; i < arr.length; i++) {

        let raingClass = 'green';
        if (arr[i].Student.rating < 50) {
            raingClass = "red";
        }
        let ImagePath = "";
        if (arr[i].Student.Image == "") {
            ImagePath = '/assets/img/user.png';
        }
        else {
            ImagePath = '/assets/php' + arr[i].Student.Image;
        }
        let location=``;
        // if(arr[i].Room == 0){
        //     location=' Студент без кімнати';
        // }
        // else if(arr[i].Hostel==0){

        // }
        console.log(arr[i].Hostel == arr[i].Section == arr[i].Room == 0 );
        usersData += `<div class="search_user_item"> 
            <span class="student_name"> ${arr[i].Student.Name}</span>
            <span class="student_un"> ${arr[i].Student.University}</span>
            <span class="student_spec"> ${arr[i].Student.Speciality}</span>
            <span class="student_rating ${raingClass}">${arr[i].Student.rating}</span>
            <span class="student_location"> ${arr[i].Hostel ==0&arr[i].Section==0&arr[i].Room== 0 ? ' Студент без кімна
ти' : arr[i].Room.Title + ', ' + arr[i].Section.Title + ', ' + arr[i].Hostel.Title + ' за адресою -
' + arr[i].Hostel.Address}</span></div>`;
    }
    item += `
        <div class="search_result_item">

```

```

        <div class="users_list">
            ${usersData}
        </div>
    </div>;
    $('<search_result .search_result_body').append(item);
}
function setRoomsInSerchResult(arr, search) {
    $('<search_result .search_result_body').empty();
    $('<search_result>h2').text('Результати пошуку за запитом - ' + search + ' :');
    $('<edit_db').fadeOut(200);
    let item = '';
    for (let i = 0; i < arr.length; i++) {
        let usersData = '';
        if (arr[i].users != 0) {
            console.log(arr[i].users);
            for (let j = 0; j < arr[i].users.length; j++) {
                let raingClass = 'green';
                if (arr[i].users[j].rating < 50) {
                    raingClass = "red";
                }
                let ImagePath = '';
                if (arr[i].users[j].Image == "") {
                    ImagePath = '/assets/img/user.png';
                }
                else {
                    ImagePath = '/assets/php' + arr[i].users[j].Image;
                }
                usersData += `<div class="search_user_item"> 
                <span class="student_name"> ${arr[i].users[j].Name}</span>
                <span class="student_un"> ${arr[i].users[j].University}</span>
                <span class="student_spec"> ${arr[i].users[j].Speciality}</span>
                <span class="student_rating ${raingClass}>${arr[i].users[j].rating}</span></div>`;
            }
        }
        else {
            usersData = `<h3 class="emptyStudents">Ця кімната вільна</h3>`
        }

        item += `
        <div class="search_result_item">
            <div class="search_result_item_name">
                ${arr[i].room.Title}, ${arr[i].sections != 0 ? arr[i].sections.Title : ''}, ${arr[i].hostel != 0 ? arr[i].hostel.Title + ' з
а адресою - ' + arr[i].hostel.Address : ''}
            </div>
            <div class="users_list">
                ${usersData}
            </div>
        </div>`;
        $('<search_result .search_result_body').append(item);
    }
}
$('<exit').on('click',function(){
    $.ajax({
        url: '/assets/php/server.php',
        type: 'POST',
        data: { exit: "1" },
    })
    .done(function (data) {
        window.location.href='../';
    });
});

```



```

    })
  })

```

- footer.css

```

.footer {
  background-color: #0A1C2B;
  position: fixed;
  left: 0;
  bottom: 0;
  width: 100%;
  padding: 12px;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}

.footer__span {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  font-size: 18px;
  color: #fff;
  padding-bottom: 9px;
}

```

- header_unreg.css

```

.header {
  background-color: #0A1C2B;
  padding: 15px;
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 100px;
}

.header__link {
  font-size: 18px;
  color: #fff;
  text-decoration: none;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

.header__logo {
  font-size: 25px;
  color: #fff;
  text-decoration: none;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

.add_student_in_room .new_student_in_room_item .students_in_room_check:hover {
  background: rgba(10, 28, 43, .3);
}

```

```
}

.add_student_in_room .new_student_in_room_item .students_in_room_check:after {
  content: '✔';
  color: white;
  position: absolute;
  top: 0;
  left: 0;
  width: 20px;
  height: 20px;
}

.add_student_in_room .new_student_in_room_item .students_in_room_check.checked {
  background: #0A1C2B !important;
}

.list_users_without_room_body_list .db_res_list_item {
  color: black;
  width: 80%;
  margin: 0 auto;
}

.list_users_without_room_body {
  color: black;
}

.list_users_without_room_body>h2 {
  text-align: center;
  margin-bottom: 50px;
}

.create_user_body {
  color: black;
  display: none;
}

.create_user_body>.input_wrap {
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  margin-left: 25px;
  position: relative;
}

.create_user_body>.input_wrap>input {
  padding: 10px 0;
  text-indent: 10px;
  border-radius: 10px;
  margin-top: 10px;
  width: 70%;
}

.create_user_body>.input_wrap>label {
  font-weight: bold;
  margin-top: 20px;
}

.create_user_body>h2 {
  text-align: center;
}
```

```
}

.image_preview {
  width: 100px;
  height: 100px;
  position: absolute;
  left: 29%;
  top: 15px;
}

.list_users_without_room_body {
  display: none;
}

.create_student {
  margin-top: 40px;
  padding: 10px;
  font-weight: bold;
  border: 2px solid black;
  cursor: pointer;
  width: 25%;
  text-align: center;
  margin: 0 auto;
}

.search{
  display:flex;
  width:30%;
  margin-left:65%;
  align-items:center;
  margin-top:25px;
}

.search>input,.search>div{
  padding:10px 0;
  border:none;
  color:black;
  border:1px solid black;
  text-align:center;
  outline:none;
}

.search>input{
  padding:13px 0;
  width:70%;
  border-right:none !important;
}

.search>div{
  width:30%;
  font-weight:bold;
  cursor:pointer;
  transition:.3s all;
}

.search>div:hover{
  background:black;
  color:white;
}

.search_result{
  display:none;
}

.search_result_item{
  color:black;
```

```

}
.search_result>h2{
  color:black;
  text-align:center;
}
.close_search_result{
  position:absolute;
  right:20px;
  width:50px;
  cursor:pointer;
}
.search_result_item_name{
  margin-left:15%;
  margin-bottom:25px;
  font-weight:bold;
}
.search_result_item{
  display: flex;
  flex-direction: column;
  align-items:flex-start;
}
.search_user_item{
  display:flex;
  width:70%;
  justify-content: flex-start;
  align-items:center;
  padding-bottom:10px;
  border-bottom:2px solid black;
  margin-top:10px;
}
.user_item_left{
  max-width:500px;
}
.user_item_left>img{
  width:100%;
}
.user_item_right{
  margin-left:25px;
}
.search_user_item>img{
  width:100px
}
.user_item_right{
  display: flex;
  flex-direction: column;
  align-items: flex-start;
}
}
.user_item_right .student_name{
  font-weight:bold;
}
}
.exit{
  cursor: pointer;
  font-weight:bold;
}
}

```

- signin.css

```
.container {
  display: flex;
  justify-content: center;
}

.signin-form {
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  padding: 25px;
  width: 300px;
  height: 200px;
}

.signin-form__section {
  display: flex;
  flex-direction: column;
}

.signin-form__submit-section {
  display: flex;
  justify-content: center;
}

.signin-form__field {
  font-family: sans-serif;
  padding: 4px 6px;
  font-family: Tahoma, Geneva, Verdana, sans-serif;
  border-color: #0A1C2B;
  border-radius: 7px;
  color: #0A1C2B;
}

.signin-form__label {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  background-color: white;
  width: min-content;
  padding: 8px 0px;
}

.signin-form__submit-button {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  padding: 7px;
  border: 2px solid #0A1C2B;
  background-color: #fff;
  border-radius: 7px;
  font-family: sans-serif;
  color: #0A1C2B;
  transform: .3s;
}

.signin-form__submit-button:hover {
  color: white;
  background-color: #0A1C2B;
}
```