АНОТАЦІЯ

Ця курсова робота спрямована на вивчення правил проектування баз даних, використання нормальних форм та надання різного доступа до окремих складових бази даних. У даній роботі розробляється автоматизована інформаційна система для вищого навчального закладу. У програмі використовуються також мови для розробки графічного інтерфейсу (React) та для написання серверу (Node JS)

Документація до курсової роботи містить:

* аналіз та опис предметної області;
* покроковий опис проектування бази даних;
* опис проектування користувацького інтерфейсу;
* опис адміністрування системи;
* детальне керівництво користувача, опис контрольних прикладів, тестової бази даних, а також тестування.

ВСТУП

Актуальність теми

У наш час дуже важливою частиною суспільного життя є отримання освіти. Кожен бажає навчатися комфортно, отримуючи всю необхідну організаційну інформацію вчасно та бути впевненим, що вона є актуальної. З метою, щоб полегшити життя студентів, викладачів та організаторів учбового процесу, створюються автоматизована інформаційна система (АІС) для вищого навчального закладу. Досить швидко, маючи доступ до мережі Інтернет або маючи встановлену десктопну версію програми можна швидко отримувати інформацію про розклад та зміни в ньому, оцінки за контрольні та екзаменаційні роботи, передивлятися борги за оплату гуртожитку тощо. З розвитком сучасних технологій подібні системи з нагадування стають більш популярними.

Метою створення курсового проекту є покращення якості організації учбового процесу у вищому навчальному закладі, а також більш комфортне отримання студентами необхідної інформації про розклад, рейтингові бали тощо. Також АІС буде використовуватися для спрощеного контроля поселення та оберту інвентарю у гуртожитках та видачі різноманітних пільг для проживаючих. БД створюється для швидкого отримання інформації про студентів, викладачів, корпуси та аудиторії, про навчальні дисципліни, проведення занять, можливість поселення в гуртожиток, наявний інвентар, пільги та оплату, а також звітів про отримані бали під час сесійного контролю для подальшого розрахунку, що впливає на отримання студентом стипендії.

Для досягнення мети у роботі вирішуються наступні задачі:

* збір даних;
* проектування актуальної бази даних, враховуючи всі обмеження цілісності даних;
* надавання різного доступу до таблиць бази даних;
* реалізація графічного інтерфейсу для зручної роботи з інформаційною систему;
* генерування звітності.

Для забезпечення надійності системи необхідно реалізувати наступні вимоги:

* цілісність та несуперечливість даних;
* достовірність даних;
* простота керування даними;
* безпека доступу до даних.

ТЕОРЕТИЧНА ЧАСТИНА

1. Аналіз та опис предметної області
2. Опис предметної області

Відповідно до предметної області система будується з врахуванням наступних особливостей:

* У ВНЗ існують декілька підрозділів (факультет, інститут тощо).
* У кожному підрозділі є декілька кафедр.
* На кожній кафедрі працюють багато викладачів.
* На кожній кафедрі є декілька навчальних програм в залежності від обраної спеціальності.
* Студент навчається в одній конкретній групі, що знаходиться лише на 1 спеціальності та кафедрі.
* Одна дисципліна може викладатися багатьма викладачами та в різних групах.
* Після вивчення дисципліни передбачається певний вид контролю (залік, екзамен, контрольна)
* За іспит кожному студенту викладач виставляє оцінку.
* Заняття проводиться конкретним викладачем, в конкретній групі, в точно зазначений час, в певній аудиторії, що розташована в корпусі, про який теж має міститися інформація.
* У студента є можливість потребувати поселення в гуртожиток, якщо він не місцевий.
* Кожен житель гуртожитку обов’язково є студентом, але студент не обов’язково є жителем гуртожитку.
* Якщо студент поселяється в гуртожиток, то додається інформація про його таланти, пільги, наміри отримувати субсидію, а також про кімнату в якій він проживає.
* Якщо студент користується пільгою, то він не може отримувати субсидію, та навпаки.
* Студент кожен місяць має проводити оплату та мати при собі квитанцію.

Після аналізу предметної області можна виділити ряд сутностей та приступити до проектування інфологічної моделі.

1. Опис вхідних даних

На вхід в систему поступають дані про студентів, викладачів, організацію навчального процесу, корпуси та аудиторії, підрозділи. кафедри, еказмени, гуртожитки, кімнати, пільги, інвентар, квитанція.

Дані про студента описуються наступною інформацією: ПІБ, потреба в гуртожитку, група. Дані про викладача описуються наступною інформацією: ПІБ та дата прийняття на роботу. Підрозділ описується типом, назвою, датою створення та керівником. Інформація про корпуси подається у вигляді набору даних про номер, адресу, кількість поверхів та аудиторій. Аудиторія описується місткістю, номером, номером корпусу, в якому вона знаходиться, а також можливістю використання комп’ютерів. Кафедра характеризується назвою, підрозділом, в якому знаходиться, а також керівником. Дані про предмет містять назву та опис. Спеціальність характеризується кодом, назвою та областю. Дані про екзамени та заліки вносяться після складання, а отже містять інформацію про предмет, тип контролю, викладача, студента, оцінку, дату та час складання. Для зберігання інформації про гуртожитки вносяться дані про номер та завідуючого. Кожна кімната містить тип, ціну, місткість, інформацію про гуртожиток,в якому знаходиться. Про жителя гуртожитка є інформація щодо його здібностей, пільг та кімнати. Дані про квитанцію містять наступні поля: дата оплати, сума, початок періоду, кінець періоду та житель гуртожитку. Інвентар описується назвою та кількістю.

1. Опис вихідних даних

Вихідні дані представлені оцінками студентів, рейтинговими списками, статистикою роботи ВНЗ.

1.2 Проектування бази даних

1.2.1 Інфорлогічна модель

1.2.1.1 Опис сутностей

Для проектування бази даних виділимо наступні базові сутності: підрозділ (факультет, інститут тощо), кафедра, викладач, спеціальність, студент, група, дисципліна, тип контролю, корпус. аудиторія, пара, іспит, гуртожиток, тип кімнати, кімната та житель гуртожитку.

1.2.1.2 Опис атрибутів

Визначимо основні атрибути базових сутностей (табл. 1.2.1.2.1). Ключові атрибути підкреслимо.

Таблиця 1.2.1.2.1 – Базові сутності та їх атрибути

|  |  |
| --- | --- |
| Сутність | Атрибути |
| Підрозділ | Ідентифікатор, тип, назва |
| Кафедра | Ідентифікатор, підрозділ, назва |
| Викладач | Ідентифікатор, ПІБ, дата прийому на роботу, звання, чи є куратором |
| Спеціальність | Ідентифікатор, код спеціальності, назва |
| Студент | Ідентифікатор, ПІБ, група, чи є жителем гуртожитку, чи отримує стипендію |
| Група | Назва, наукова пропозиція, кількість студентів, куратор |
| Дисципліна | Ідентифікатор, назва |
| Тип контролю | Ідентифікатор, вид |
| Корпус | Номер корпусу, адреса, кількість аудиторій, кількість поверхів |
| Аудиторія | Ідентифікатор, корпус, кількість осіб, яку вміщує |
| Пара | Ідентифікатор, дисципліна, викладач, група, тип контролю, час |
| Іспит | Ідентифікатор, студент, дисципліна, викладач, кафедра, група, тип контролю, оцінка, аудиторія, час |
| Гуртожиток | Номер, керівник, адреса, кількість кімнат, місткість |
| Кімната | Ідентифікатор, номер гуртожитку, місткість, ціна |
| Житель гуртожитку | Ідентифікатор, студент, кімната, здібності, пільга, субсидія |

1.2.1.3 Опис зв’язків

Визначимо зв’язки між базовими сутностями та охарактеризуємо їх (табл. 1.2.1.3.1)

Таблиця 1.2.1.3.1 – Зв’язки між сутностями

|  |  |
| --- | --- |
| Зв’язок | Атрибути |
| Належність кафедри | Підрозділ, Кафедра |
| Працівник на кафедрі | Ідентифікатор, кафедра, викладач |
| Наукова пропозиція | Ідентифікатор, спеціальність, кафедра |
| Викладання дисципліни | Ідентифікатор, дисципліна, викладач, група, тип контролю |
| Кураторство | Група, працівник кафедри |
| Навчання в групі | Група, студент |
| Належність аудиторії корпусу | Корпус, аудиторія |
| Пара | Ідентифікатор, викладання дисципліни, аудиторія, час |
| Проживання в гуртожитку | Студент, житель гуртожитку |
| Складання іспиту | Ідентифікатор, студент, викладання дисципліни, оцінка, аудиторія, час |
| Проживання в кімнаті | Житель гуртожитку, кімната |
| Розташування кімнати | Гуртожиток, кімната |

Зазначимо наступні обмеження та дозволи предметної області:

* Ключові поля мають бути унікальними.
* Житель гуртожитку обов’язково має бути студентом цього ВНЗ.
* Оцінкою студента після іспиту є оцінка, набрана за семестр, в разі якщо він не з’явився, або загальний бал. Обов’язково має бути число після проведення іспиту.
* Знижка на проживання в гуртожитку може бути лише одна, або за наявності пільги, або за намірів отримувати субсидію.

1.2.1.4 діаграма “сутність-зв’язок”

На основі виділених сутностей, атрибутів та зв’язків побудуємо ER-діаграму. (рис. 1.2.1.4.1)

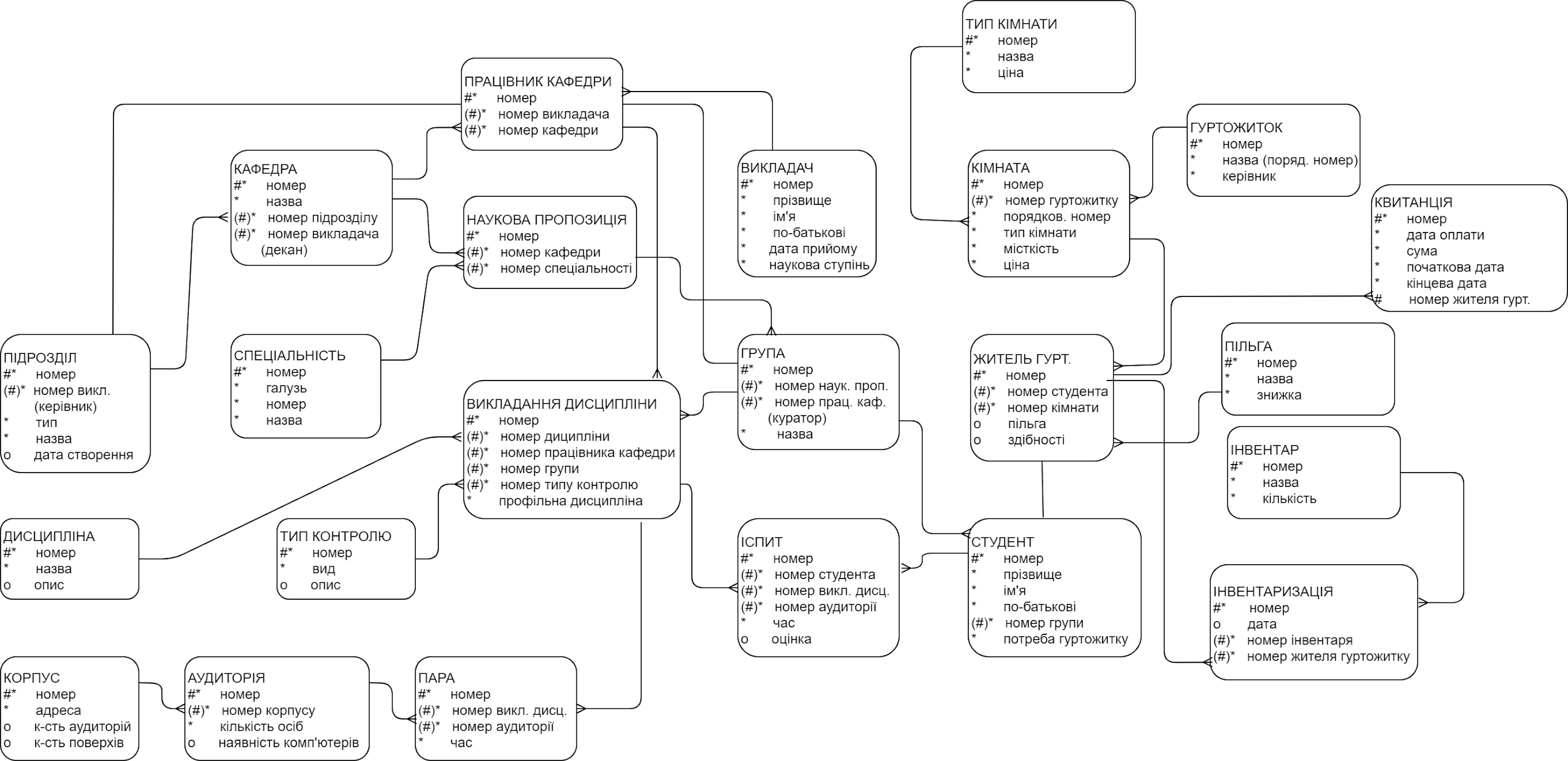


рис. 1.2.1.4.1 - Діаграма “сутність-зв’язок”

1.2.2 Нормалізація таблиць при проектуванні бази даних

При визначенні основних сутностей та їх атрибутів ми використовували нормалізацію. Для приведення бази даних до першої нормальної форми ми кожній таблиці надали ключові поля та намагалися правильно визначити неключові поля для усунення повторень груп. Дані, що зустрічалися декілька разів ми виносили в окремі таблиці. Для розв’язання зв’язків “багато-до-багатьох” ми додавали нові зв’зуючі таблиці. Наприклад: Викладання дисципліни, Працівник кафедри, Іспит.

1.2.3 Даталогічна модель бази даних

Визначимо типи даних для кожного поля сутності та занесемо дані в таблиці.

Таблиця 1.2.3.1 - DEPARTMENT (ПІДРОЗДІЛ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| HEAD\_ID | FK | NOT NULL | ACADEMIC | ID | NUMBER | 10 |
| TYPE |  | NOT NULL |  |  | VARCHAR2 | 15 |
| TITLE |  | NOT NULL |  |  | VARCHAR2 | 50 |
| CREATION\_  DATE |  |  |  |  | DATE |  |

Таблиця 1.2.3.2 - CATHEDRA (КАФЕДРА)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| TITLE |  | NOT NULL |  |  | VARCHAR2 | 50 |
| DEPARTMENT\_  ID | FK | NOT NULL | DEPARTMENT | ID | NUMBER | 10 |
| DEAN\_ID | FK | NOT NULL | ACADEMIC | ID | NUMBER | 10 |

Таблиця 1.2.3.3 - ACADEMIC (ВИКЛАДАЧ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | ID |
| FIRST NAME |  | NOT NULL |  |  | VARCHAR2 | 20 |
| MIDDLE NAME |  | NOT NULL |  |  | VARCHAR2 | 20 |
| LAST NAME |  | NOT NULL |  |  | VARCHAR2 | 20 |
| RECRUITMENT  \_DATE |  | NOT NULL |  |  | DATE |  |

Таблиця 1.2.3.4 - SPECIALTY (СПЕЦІАЛЬНІСТЬ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| AREA |  | NOT NULL |  |  | NUMBER | 2 |
| CODE |  | NOT NULL |  |  | NUMBER | 2 |
| TITLE |  | NOT NULL |  |  | VARCHAR2 | 50 |

Таблиця 1.2.3.5 - DISCIPLINE (ДИСЦИПЛІНА)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| TITLE |  | NOT NULL |  |  | VARCHAR2 | 50 |
| DESCRIPTION |  |  |  |  | TEXT |  |

Таблиця 1.2.3.6 - CONTROL\_TYPE (ТИП КОНТРОЛЮ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| TYPE |  | NOT NULL |  |  | VARCHAR2 | 20 |
| DESCRIPTION |  |  |  |  | TEXT |  |

Таблиця 1.2.3.7 - CATHEDRA\_EMPLOYEE (ПРАЦІВНИК КАФЕДРИ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| ACADEMIC\_ID | FK | NOT NULL | ACADEMIC | ID | NUMBER | 10 |
| CATHEDRA\_ID | FK | NOT NULL | CATHEDRA | ID | NUMBER | 10 |

Таблиця 1.2.3.8 - НАУКОВА ПРОПОЗИЦІЯ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| CATHEDRA\_ID | FK | NOT NULL | CATHEDRA | ID | NUMBER | 10 |
| SPECIALTY\_ID | FK | NOT NULL | SPECIALTY | ID | NUMBER | 10 |

Таблиця 1.2.3.9 - GROUP (ГРУПА)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| TITLE |  | NOT NULL |  |  | VARCHAR2 | 5 |
| НАУКОВА ПРОПОЗИЦІЯ  \_ID | FK | NOT NULL | НАУКОВА ПРОПОЗИЦІЯ | ID | NUMBER | 10 |
| CURATOR\_ID | FK | NOT NULL | CATHEDRA | ID | NUMBER | 10 |

Таблиця 1.2.3.10 - BUILDING (КОРПУС)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| ADDRESS |  | NOT NULL |  |  | VARCHAR2 | 30 |
| AUDITORIUMS  \_AMOUNT |  |  |  |  | NUMBER | 3 |
| FLOORS\_  AMOUNT |  |  |  |  | NUMBER | 2 |

Таблиця 1.2.3.11 - AUDITORIUM (АУДИТОРІЯ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| BUILDING\_ID | FK | NOT NULL | BUILDING | ID | NUMBER | 10 |
| CAPACITY |  | NOT NULL |  |  | NUMBER | 3 |
| COMPUTERS\_AVAILABILITY |  |  |  |  | CHAR | 1 |

Таблиця 1.2.3.12 - ВИКЛАДАННЯ ДИСЦИПЛІНИ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| DISCIPLINE\_ID | FK | NOT NULL | DISCIPLINE | ID | NUMBER | 10 |
| CATHEDRA\_EMPLOYEE\_ID | FK | NOT NULL | CATHEDRA\_  EMPLOYEE | ID | NUMBER | 10 |
| GROUP\_ID | FK | NOT NULL | GROUP | ID | NUMBER | 10 |
| CONTROL\_  TYPE\_ID | FK | NOT NULL | CONTROL\_  TYPE | ID | NUMBER | 10 |
| BASIC\_  DISCIPLINE |  | NOT NULL |  |  | CHAR | 1 |

Таблиця 1.2.3.13 LESSON (ПАРА)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| ВИКЛАДАННЯ ДИСЦИПЛІНИ  \_ID | FK | NOT NULL | ВИКЛАДАННЯ ДИСЦИПЛІНИ | ID | NUMBER | 10 |
| AUDITORIUM  \_ID | FK | NOT NULL | AUDITORIUM | ID | NUMBER | 10 |
| TIME |  | NOT NULL |  |  | DATE |  |

Таблиця 1.2.3.14 - EXAM (ІСПИТ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| STUDENT\_ID | FK | NOT NULL | STUDENT | ID | NUMBER | 10 |
| ВИКЛ ДИСЦ\_ID | FK | NOT NULL | ВИКЛ ДИСЦ | ID | NUMBER | 10 |
| AUDITORIUM\_ID | FK | NOT NULL | AUDITORIUM | ID | NUMBER | 10 |
| TIME |  | NOT NULL |  |  | DATE |  |
| MARK |  |  |  |  | NUMBER | 2 |

Таблиця 1.2.3.15 - STUDENT (СТУДЕНТ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| FIRST NAME |  | NOT NULL |  |  | VARCHAR2 | 20 |
| MIDDLE NAME |  | NOT NULL |  |  | VARCHAR2 | 20 |
| LAST NAME |  | NOT NULL |  |  | VARCHAR2 | 20 |
| GROUP\_ID | FK | NOT NULL | GROUP | ID | NUMBER | 10 |
| HOSTEL\_NEED |  | NOT NULL |  |  | CHAR | 1 |

Таблиця 1.2.3.16 - HOSTEL (ГУРТОЖИТОК)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| TITLE |  | NOT NULL |  |  | VARCHAR2 | 10 |
| HEAD |  | NOT NULL |  |  | VARCHAR2 | 50 |

Таблиця 1.2.3.17 - ROOM (КІМНАТА)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| HOSTEL\_ID | FK | NOT NULL | HOSTEL | ID | NUMBER | 10 |
| NUMBER |  | NOT NULL |  |  | NUMBER | 3 |
| CAPACITY |  | NOT NULL |  |  | NUMBER | 1 |
| PRICE |  | NOT NULL |  |  | NUMBER | 4 |

Таблиця 1.2.3.18 - HOSTEL\_RESIDENT (ЖИТЕЛЬ ГУРТОЖИТКУ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування стовпця | Тип ключа | NOT NULL | Таблиця FK | Стовпець FK | Тип даних | Довжина |
| ID | PK | NOT NULL |  |  | NUMBER | 10 |
| STUDENT\_ID | FK | NOT NULL | STUDENT | ID | NUMBER | 10 |
| ROOM\_ID | FK | NOT NULL | ROOM | ID | NUMBER | 10 |
| PRIVILEGE |  |  |  |  | NUMBER | 3 |
| SUBSIDY |  |  |  |  | NUMBER | 2 |
| ABILITIES |  |  |  |  | TEXT |  |

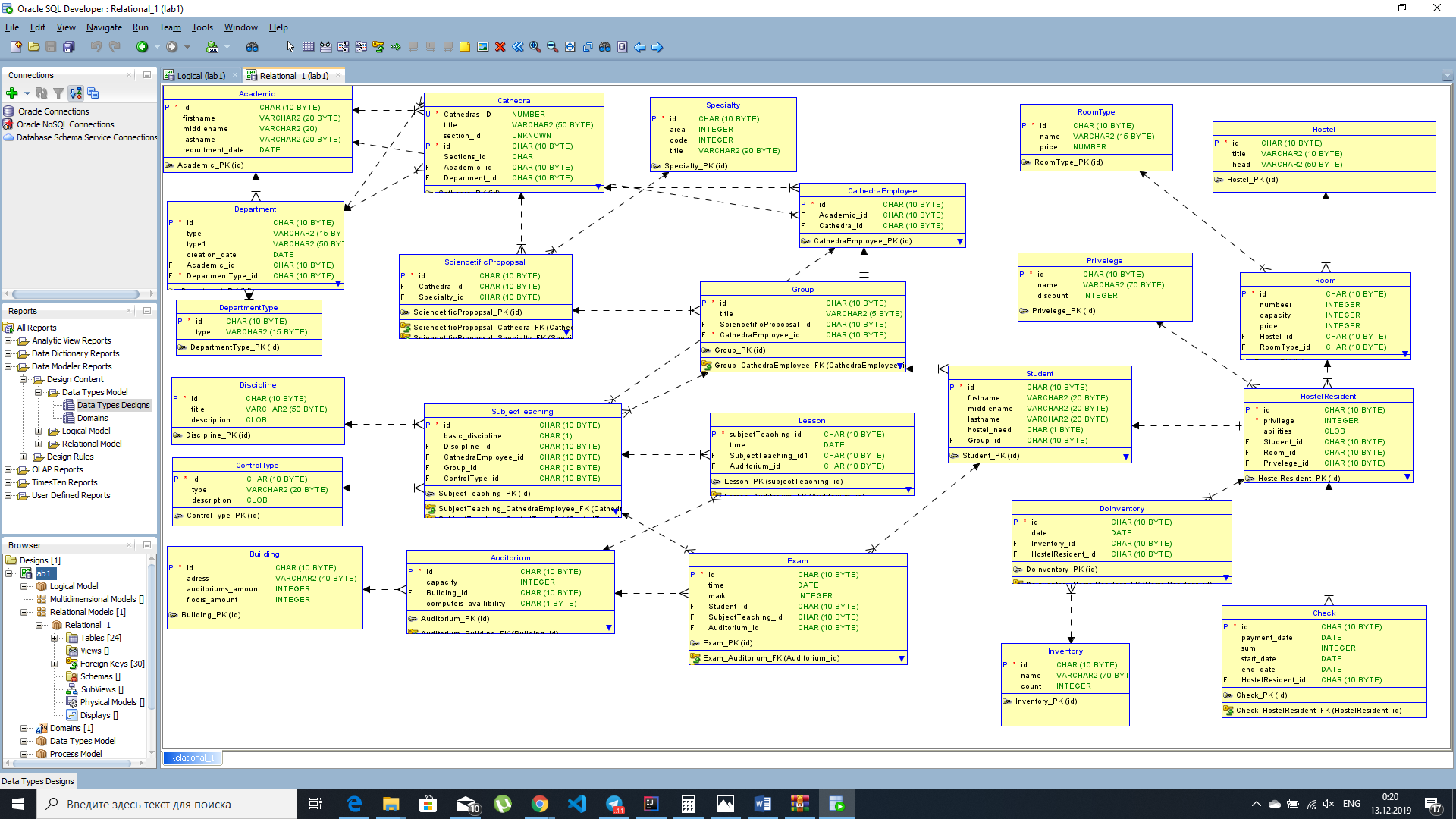


рис. 1.2.3.1 - Даталогічна модель

1.3 Проектування користувацького інтерфейсу

Для всіх трьох підсистем (деканату, студентства та студмістечка) має бути одна сторінка з пропозицією авторизуватися, тобто ввести логін та пароль. Вони видаються кожному працівнику навчального закладу або студенту адміністрацією. Після авторизації в залежності від посади, кожен співробітник переходить на окрему сторінку.

Вгорі кожної сторінки має знаходитися “виповзаюче” меню з однаковими функціями:

* змінити пароль;
* керувати профілем;
* інформація;
* вихід з системи.

Керування профілем включає в себе зміну деякої особистої інформації (ім’я, електронна пошта, наукові інтереси, кредо) та представлення незмінних полів (підрозділ, позиція).

Інформація містить наступні пункти:

* інструкція користувача;
* правила використання інформації сайту;
* інформація про систему;
* дошка оголошень;
* контактні дані.

В хедері має міститися логотип ВНЗ, що є гіперпосиланням на головну сторінку.

В футері має знаходитися контактна інформація та посилання на соціальні мережі.

В головній секції розташоване привітання.

Для кожного користувача існує додатковий набір функцій.

*Деканат*

|  |  |  |
| --- | --- | --- |
| Розклад занять/сесії | додати розклад | для групи |
| для викладача |
| змінити розклад | для групи |
| для викладача |
| видалити розклад | за предметом |
| за групою |
| за викладачем |
| переглянути розклад | для групи |
| для викладача |
| Результати контролю | Перегляд заліків/ екзаменів | студента |
| групи |
| викладача |
| з предмета |
| Контактна інформація | Перегляд/додавання | студентів |
| викладачів |

*Викладач*

* перегляд розкладу;
* перегляд списку груп;

*Студент*

* перегляд розкладу;
* перегляд інформації про викладачів;
* перегляд балів (поточних з предмету, за заліки/сесію, середнього).

*Працівник студмістечка*

* перегляд інформації про гуртожитки;
* перегляд список студентів/відфільтрованих за гуртожитками;
* перегляд боржників з сумою боргу;
* зміна цін на проживання за типом кімнати та гуртожитком;
* перегляд наявного/виданого інвентаря студентам певного гуртожитку/ конкретному студенту/ кімнаті;
* поселення/виселення студентів;
* формування квитанції, враховуючи пільги;
* додавання пільг до облікового запису студента;
* перегляд контактної інформації про жителів гуртожитку.

1.4 Адміністрування системи

Адміністрування системи відбувається за участі працівників деканату та студмістечка. Методист може додавати/змінювати/видаляти розклад, додавати нових студентів, а також вносити і коригувати інформацію про підрозділи і їх працівників.

Працівник студмістечка може додавати, видаляти студента з бази даних, вносити, коригувати, видаляти інформацію про інвентар та кімнати, а також виставляти рахунки жителям гуртожитків.

ПРАКТИЧНА ЧАСТИНА

2.1 Структура програмного забезпечення

2.1.1 Створення бази даних

Для створення бази даних ми обрали СУБД PostgreSQL. Порівняємо її з Oracle. (табл.2.1.1.1)

Таблиця 2.1.1.1 - Порівняння СУБД

|  |  |  |
| --- | --- | --- |
|  | Oracle | PostgreSQL |
| Ціна | Є високою, за кожну додаткову функцію необхідно платити. | Всі доступні можливості ми можемо отримувати безкоштовно, оскільки ця СУБД є open-source. |
| Підтримка | Підтримка клієнтів для БД не є безкоштовною | Підтримка клієнтів безкоштовна, але знадобиться час для вирішення цієї проблеми, оскільки вона має бути вирішена спільнотою розробника. Є можливість користуватися платною послугою підтримки професіоналами, але це дешевше, ніж у порівнюваної СУБД. |
| Продуктивність | Вища завдяки технічній перевазі. Забезпечує більше транзакцій в секунду, ніж PostgreSQL. | Менша, оскільки вона забезпечує меншу кількість транзакцій. |
| Безпека | Має більш високу безпеку, але нам необхідно купувати частину видань корпорації, які мають функції, що захищають БД. | Має хорошо підтримку, але не таку просунуту як БД Oracle. |
| Масштабованість | Для проектів з високим навантаженням потрібно придбати корпоративну версію | Підтримує масштабованість для безкоштовних розширень витрат |
| Оновлення | Оновлюється раз на 3-4 роки, щоб триматися на верхніх позиціях на ринку | Оновлюється раз на 4-5 років, при цьому додаються нові сучасні та більш зручні особливості. |
| Обробка великої кількості даних | Enterprise Edition обробляє великий обсяг даних більш ефективно, ніж PostgreSQL на основі рівних умов та типів машин | Ефективно опрацьовує великі обсяги даних, що підвищує продуктивність на 10-30 сторінок на машинах з великими обсягами пам’яті. |

2.1.2 Організація вибірки інформації з бази даних

Користувачі можуть виконувати наступні дії з АІС з врахуванням обмеження доступу деяких користувачів до окремих таблиць БД:

Деканат:

* додавання розкладу групі;
* додавання розкладу викладачу;
* редагування розкладу;
* перегляд розкладу для групи, викладача, аудиторії;
* перегляд контактної інформації студентів та працівників факультету;
* додавання нового студента, працівника до БД;
* перегляд результатів заліку/екзамену для конкретного студент
* перегляд результатів заліку/екзамену для групи;
* перегляд результатів заліку/екзамену для викладача;
* перегляд результатів заліку/екзамену для конкретного предмету;
* перегляд інформації про викладачів;
* перегляд списку студентів;
* перегляд списку студентів конкретної групи.

Викладач:

* перегляд списка студентів групи;
* перегляд розкладу конкретної групи;
* перегляд розкладу для викладача.

Студент:

* перегляд поточних оцінок/балів за залік або екзамен;
* перегляд розкладу для групи;
* перегляд розкладу для викладачів;
* перегляд інформації про гуртожитки;
* перегляд інформації про викладачів.

Працівник студмістечка:

* додавання нового студента;
* перегляд списку жителів;
* перегляд списку жителів за кімнатою;
* додавання квитанції;
* перегляд боржників з сумою боргу;
* перегляд наявного у гуртожитку інвентаря;
* перегляд виданого інвентаря студентам;
* перегляд виданого інвентаря конкретному студенту;
* видалення жителя гуртожитку;
* перегляд контактної інформації про студентів;
* зміна цін на проживання;

2.2 Опис програм

## 2.2.1 Виклик і завантаження

Програма є веб-сайтом ВНЗ, отже для доступу до нього необхідно мати доступ до мережі Інтернет. При переході на сайт з’являється головна сторінка з привітанням.

## 2.2.2 Вхідні дані

Вхідні дані в програму передаються за допомогою клавіатури (користувачу пропонується ввести дані в форму). Доступ до кнопок, що розташовані на сайті відбувається через ЛКМ.

## 2.2.3 Вихідні дані

Всі дані програми виводяться через екран та запис в БД. Всі дані, що збережені в БД мають відображатися у відповідних формах, якщо особа має право доступу до них. Повідомлення про некоректний ввід відображається у вигляді окремого вікна, де вказано тип помилки. тобто що було введено неправильно. Також реалізовані повідомлення про відсутність бажаних даних у БД.

2.3 Керівництво користувача

Програма має назву “Campus”. Вона призначена переважно для користування з інформативною та обліковою метою. Але також вона має бути корисною для зберігання та непошкодження усіх важливих даних, які є важливою складовою робочого та навчального процесу у ВНЗ.

Після завантаження сторінки користувач побачить вікно з пропозицією авторизуватися. Потім натискаючи на кнопки з відповідними написами можна отримати необхідну інформацію або за можливістю. користуючись підказками на екрані додати дані. При спробі ввести некоректні дані програма видать повідомлення про помилку та вкаже, що конкретно було введено неправильно. У разі відсутності шуканих даних користувач побачить інформацію про це.

2.3.1. Умови застосування програми

Для функціонування цієї програми не потрібно додаткового програмного забезпечення. Потрібен доступ до мережі Інтернет, ПК з мінімальними параметрами, планшет або смартфон. Необхідні периферійні пристрої такі як монітор, клавіатура і комп’ютерна миша.

2.3.2 Послідовність дій користування програмою

Підготовки вхідних даних та налаштувань перед початком роботи проводити не потрібно.

Перехід від однієї точки діалогу до іншої відбувається за допомогою кнопок, що розташовані на сторінках веб-сайту. При натисканні відбувається перехід з одного сторінки на іншу, потім деякі кнопки зникають, і з’являється нові, а в деяких лишаються з додаванням нових. Призначення режимів роботи додатку зрозумілі з назв та символів, зображених на кнопках та в якості логотипа.

2.4. Вимоги до апаратних та програмних засобів

Оскільки програма не потребує великих обчислювальних можливостей, то вона не має особливих вимог для коректного запуску: мінімальні характеристики ПК – процесор Intel Celeron, 2 ГБ оперативної пам’яті; операційна система 7, 8, 10 або мінімальні характеристики портативного пристрою.

Для доступу до додатку необхідно мати доступ до мережі Інтернет, а також встановити браузер на комп’ютер або портативний пристрій.

ВИПРОБУВАННЯ СИСТЕМИ

## 3.1 Опис контрольних прикладів

1. Додати до БД нового жителя гуртожитка студента Петренко Віктора Олеговича.
   1. Випадок такий студент вже живе в гуртожитку (будь-якому): користувач побачить повідомлення, про те, що цього студента неможливо додати, оскільки він вже живе в гуртожитку
   2. Випадок, коли студента з такими даними не існує, дані про нього мають додатися до БД.
2. Додати до БД користувача з некоректними даними, наприклад з прізвища, що починається з цифри. Користувач побачить повідомлення про те, що дані введені некоректно.
3. Додати в квитанцію на оплату гуртожитку в поле “Сума” некоректні дані. Наприклад, літера. Користувач побачить повідомлення про некоректність даних.
4. Зробити оплату для студента Давиденко Тетяна Віталіївна (такого користувача не існує в БД) неможливо. Користувач отримує повідомлення про помилку.
5. Змінити корекні дані про користувача чи розклад на некоректні (повторювані або з неправильним форматом вводу)
6. Видалити студента Петренко Віктора Олеговича (Студента з такими даними немає вже бути в базі даних)
7. Виконати вибірку даних
   1. Зробити запит на вибір студентів групи ІТ-72
   2. Зробити запит на вибір студентів, що здійснили оплату
   3. Запит на отримання інформації про конкретного викладача
   4. Запит на отримання інформації про розклад.

3.2 Опис тестової БД

Для тестування розробленої нами БД на коректність, ми додали до основних таблиць декілька записів, з якими потім робили маніпулювання, виконуючи запити та визначали коректність даних.

Таблиця 3.1.1 - Таблиця з даними “Викладачі”

|  |  |  |  |
| --- | --- | --- | --- |
| Ім’я | По-батькові | Прізвище | Дата прийняття на роботу |
| Ярослав | Ігорович | Корнага | 10.05.1965 |
| Олег | Іванович | Лісовиченко | 10.05.1965 |
| Михайло | Володимирович | Кушлаба | 10.08.1969 |
| Ірина | Костянтинівна | Покулита | 10.05.1965 |
| Аліна | Дмитрівна | Курочкіна | 10.05.1965 |
| Наталя | Станіславівна | Корнієнко | 10.05.1965 |
| Генадій | Вікторович | Рябцев | 10.05.1965 |
| Артем | Петрович | Заворикін | 10.05.1965 |
| Владислав | Олегович | Мамедов | 25.10.1987 |
| Валерій | Михайлович | Ніколенко | 15.07.1999 |

Таблиця 3.1.2 - Таблиця з даними “Тип підрозділу”

|  |
| --- |
| Тип |
| факультет |
| інститут |

Таблиця 31.3 - Таблиця з даними “Підрозділ”

|  |  |  |
| --- | --- | --- |
| Назва | Дата створення | Підрозділу |
| Факультет інформатики та обчислювальної техніка | 10.05.1965 | факультет |
| Інститут прикладного системного аналізу | 10.05.1965 | інститут |
| Факультет менеджменту та маркетингу | 10.08.1969 | факультет |
| Теплоенергетичний факультет | 10.05.1965 | факультет |
| Зварювальний факультет | 10.05.1965 | факультет |
| Факультет прикладної математики | 10.05.1965 | факультет |
| Факультет соціології та права | 10.05.1965 | факультет |
| Факультет лінгвістики | 10.05.1965 | факультет |
| Фізико-технічний інститут | 25.10.1987 | факультет |
| Радіотехнічний факультет | 15.07.1999 | факультет |

Таблиця 3.1.4 - Таблиця з даними “Корпуси”

|  |  |  |  |
| --- | --- | --- | --- |
| Адреса | Кількість аудиторій | Кількість поверхів | Номер |
| просп. Перемоги, 1 | 100 | 5 | 1 |
| просп. Перемоги, 3 | 250 | 7 | 2 |
| просп. Перемоги, 5 | 150 | 6 | 3 |
| вул. Академіка Янгеля, 10 | 500 | 12 | 7 |
| вул. Металістів, 15А | 60 | 4 | 12 |
| вул. Виборзька, 123Б | 120 | 5 | 8 |
| просп. Перемоги, 110 | 250 | 8 | 4 |
| просп. Перемоги, 220 | 300 | 12 | 9 |
| вул. Академіка Янгеля, 35 | 400 | 10 | 11 |
| вул. Металістів, 21В | 110 | 5 | 19 |

Таблиця 3.1.5 - Таблиця з даними “Аудиторії”

|  |  |  |  |
| --- | --- | --- | --- |
| Місткість | Корпус | Наявність комп’ютерів | Номер |
| 35 | 1 | так | 201 |
| 110 | 2 | ні | 301 |
| 50 | 3 | ні | 438 |
| 27 | 7 | так | 415 |
| 300 | 9 | ні | 418 |
| 100 | 12 | ні | 402 |
| 65 | 11 | ні | 407 |
| 20 | 18 | так | 432 |
| 15 | 18 | так | 538 |
| 18 | 18 | так | 339 |

Таблиця 3.1.6 - Таблиця з даними “Кафедри”

|  |  |  |
| --- | --- | --- |
| Назва | Підрозділ | Керівник |
| Кафедра технічної кібернетики | Факультет інформатики та обчислювальної техніка | Ярослав Корнага |
| Кафедра системного проектування | Інститут прикладного системного аналізу | Олег Лісовиченко |
| Кафедра менеджменту | Факультет менеджменту та маркетингу | Михайло Кушлаба |
| Кафедра теоретичної та промислової теплотехніки | Теплоенергетичний факультет | Ірина Покулита |
| Кафедра інженерії поверхні | Зварювальний факультет | Аліна Курочкіна |
| Кафедра прикладної математики | Факультет прикладної математики | Наталя Корнієнко |
| Кафедра соціології | Факультет соціології та права | Генадій Рябцев |
| Кафедра української мови | Факультет лінгвістики | Михайло Заворикін |
| Кафедра інформаційної безпеки | Фізико-технічний інститут | Владислав Мамедов |
| Кафедра теоретичних основ радіотехніки | Радіотехнічний факультет | Валерій Ніколенко |

Таблиця 3.1.7 - Таблиця з даними “Предмети”

|  |  |
| --- | --- |
| Назва | Опис |
| Бази даних | Вивчення базових знань про бази даних, правила їх проектування, запити. Виконання практичних робіт з інфологічного, даталогічно проектування, нормалізації. Фізична реалізація у вибраній студентом СУБД та написання запитів для неї |
| Основи програмування | Дисципліна для отримання базових знань в області програмування |
| Основи менеджменту | Дисципліна для отримання базових знань в області менеджменту |
| Основи теплотехніки | Дисципліна для отримання базових знань в області теплотехніки |
| Основи зварювання | Дисципліна для отримання базових знань в області зварювання |
| Основи прикладної математики | Дисципліна для отримання базових знань в області прикладної математики |
| Основи права | Дисципліна для отримання базових знань в області права |
| Українська мова | Дисципліна для отримання базових знань з української мови |
| Основи інформаційної безпеки | Дисципліна для отримання базових знань в області інформаційної безпеки |
| Основи радіотехніки | Дисципліна для отримання базових знань в області радіотехніки |

Таблиця 3.1.8 - Таблиця з даними “Спеціальності”

|  |  |  |
| --- | --- | --- |
| Область | Код | Назва |
| 12 | 1 | Інженерія програмного забезпечення |
| 12 | 2 | Комп’ютерні науки |
| 07 | 3 | Менеджмент |
| 14 | 4 | Теплоенергетика |
| 13 | 1 | Прикладна механіка |
| 11 | 3 | Прикладна математика |
| 08 | 1 | Право |
| 03 | 5 | Філологія |
| 12 | 5 | Кібербезпека |
| 17 | 2 | Телекомунікації та радіотехніка |

Таблиця 3.1.9 - Таблиця з даними “Наукові пропозиції”

|  |  |
| --- | --- |
| Кафедра | Спеціальність (область-код) |
| Кафедра технічної кібернетики | 12-1 |
| Кафедра системного проектування | 12-2 |
| Кафедра менеджменту | 07-3 |
| Кафедра теоретичної та промислової теплотехніки | 14-4 |
| Кафедра інженерії поверхні | 13-1 |
| Кафедра прикладної математики | 11-3 |
| Кафедра соціології | 08-1 |
| Кафедра української мови | 03-5 |
| Кафедра інформаційної безпеки | 12-5 |
| Кафедра теоретичних основ радіотехніки | 17-2 |

Таблиця 3.1.10 - Таблиця з даними “Групи”

|  |  |  |
| --- | --- | --- |
| Назва | Наукова пропозиція | Куратор |
| ІТ-72 | Кафедра технічної кібернетики 12-1 | Ярослав Корнага |
| СА-71 | Кафедра системного проектування 12-2 | Олег Лісовиченко |
| ММ-73 | Кафедра менеджменту 07-3 | Михайло Кушлаба |
| ТФ-74 | Кафедра теоретичної та промислової теплотехніки 14-4 | Ірина Покулита |
| ЗФ-71 | Кафедра інженерії поверхні 13-1 | Аліна Курочкіна |
| ПМ-72 | Кафедра прикладної математики 11-3 | Наталя Корнієнко |
| СП-83 | Кафедра соціології 08-1 | Генадій Рябцев |
| ФЛ-64 | Кафедра української мови 03-5 | Михайло Заворикін |
| ФТ-71 | Кафедра інформаційної безпеки 12-5 | Владислав Мамедов |
| РТ-72 | Кафедра теоретичних основ радіотехніки 17-2 | Валерій Ніколенко |

Таблиця 3.1.11 - Таблиця з даними “Типи контролю”

|  |  |
| --- | --- |
| Тип | Опис |
| Іспит | Іспит - це форма контролю роботи студентів, яка може являти собою письмове завдання, усне опитування, практичне завдання або їх комбінацію. |
| Екзамен | Екзамен - це форма контролю роботи студентів, яка може являти собою письмове завдання, усне опитування, практичне завдання або їх комбінацію та на відміну від іспиту має більший вплив на успішність студента та рішення щодо отримання стипендії. Екзамен може проводитися лише з профільних предметів |
| Курсова | Курсова - це форма контролю роботи студентів, яка завжди являє собою практичне завдання з обов`язковим оформленням документації. Оцінка за курсову виставляється так само як і за інші типи контролю |

Таблиця 3.1.12 - Таблиця з даними “Студенти”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ім’я | По-батькові | Прізвище | Потреба в гуртожитку | Група |
| Хана | Ель-Дін | Халіл | так | ІТ-72 |
| Олег | Віталійович | Рикун | так | СА-71 |
| Назар | Олегович | Мартинюк | так | ММ-73 |
| Ганна | Віталіївна | Руденко | так | ММ-73 |
| Назар | Богданович | Федоренко | ні | ЗФ-71 |
| Юлія | Петрівна | Новак | так | ПМ-72 |
| Андрій | Андрійович | Рудий | так | СП-83 |
| Марія | Георгіївна | Дідович | так | ФЛ-64 |
| Святослав | Костянтинович | Кавун | так | ФТ-71 |
| Кирило | Семенович | Зікратий | так | ММ-73 |

ВИСНОВКИ

Під час виконання курсової роботи був розроблений додаток «Campus» розрахований на користувачів, що працюють або навчаються у ВНЗ. Ця атвоматизована інформаційна система призначена для організації учбового процесу, внесення, зміни та поширення актуальної інформації між працівниками та студентами. Усі дані відображаються коректно, з відповідними підписами, що дозволяє користувачу швидко розібратися з програмою без додаткової підготовки.

Правильний розподіл БД на таблиці та додатку на компоненти дозволить у майбутньому вдосконалювати програму, додаючи нові типи користувачів, а також розширювати функціонал. Оскільки програма доступна через мережу Інтернет, то вона може вільно поширюватися за допомогою гіперпосилань.

ДОДАТОК А

А.1. Скрипти на створення таблиць

CREATE TABLE academic

(

id serial PRIMARY KEY,

firstname VARCHAR(20) NOT NULL,

middlename VARCHAR(20) NOT NULL,

lastname VARCHAR(20) NOT NULL,

recruitment\_date DATE

);

CREATE TABLE departmenttype

(

id serial PRIMARY KEY,

type VARCHAR(15) NOT NULL

);

CREATE TABLE department

(

id serial PRIMARY KEY,

title VARCHAR(50) NOT NULL,

creation\_date DATE,

head\_id INTEGER REFERENCES academic(id) NOT NULL,

departmenttype\_id INTEGER REFERENCES departmenttype(id) NOT NULL

);

CREATE TABLE building

(

id serial PRIMARY KEY,

adress VARCHAR(40) NOT NULL,

auditoriums\_amount INTEGER,

number INTEGER NOT NULL,

floors\_amount INTEGER

);

CREATE TABLE auditorium

(

id serial PRIMARY KEY,

capacity INTEGER NOT NULL,

number INTEGER NOT NULL,

building\_id INTEGER REFERENCES building(id) NOT NULL,

computers\_availibility CHAR(1)

);

CREATE TABLE cathedra

(

id serial PRIMARY KEY,

title VARCHAR(50) NOT NULL,

department\_id INTEGER REFERENCES department(id) NOT NULL,

dean\_id INTEGER REFERENCES academic(id) NOT NULL

);

CREATE TABLE cathedraemployee

(

id serial PRIMARY KEY,

academic\_id INTEGER REFERENCES academic(id) NOT NULL,

cathedra\_id INTEGER REFERENCES cathedra(id) NOT NULL

);

CREATE TABLE discipline

(

id serial PRIMARY KEY,

title VARCHAR(50) NOT NULL,

description TEXT

);

CREATE TABLE specialty

(

id serial PRIMARY KEY,

area INTEGER NOT NULL,

code INTEGER NOT NULL,

title VARCHAR(90) NOT NULL

);

CREATE TABLE scientificproposal

(

id serial PRIMARY KEY,

cathedra\_id INTEGER REFERENCES cathedra(id) NOT NULL,

specialty\_id INTEGER REFERENCES specialty(id) NOT NULL

);

CREATE TABLE class

(

id serial PRIMARY KEY,

title VARCHAR(5) NOT NULL,

scientificproposal\_id INTEGER REFERENCES scientificproposal(id) NOT NULL,

curator\_id INTEGER REFERENCES cathedraemployee(id) NOT NULL

);

CREATE TABLE controltype

(

id serial PRIMARY KEY,

type VARCHAR(80) NOT NULL,

description TEXT

);

CREATE TABLE subjectteaching

(

id serial PRIMARY KEY,

basic\_discipline CHAR(1),

discipline\_id INTEGER REFERENCES discipline(id) NOT NULL,

cathedraemployee\_id INTEGER REFERENCES cathedraemployee(id) NOT NULL,

class\_id INTEGER REFERENCES class(id) NOT NULL,

controltype\_id INTEGER REFERENCES controltype(id) NOT NULL

);

CREATE TABLE student

(

id serial PRIMARY KEY,

firstname VARCHAR(20) NOT NULL,

middlename VARCHAR(20) NOT NULL,

lastname VARCHAR(20) NOT NULL,

hostel\_need CHAR(1),

class\_id INTEGER REFERENCES class(id) NOT NULL

);

CREATE TABLE exam

(

id serial PRIMARY KEY,

time DATE NOT NULL,

mark INTEGER,

student\_id INTEGER REFERENCES student(id) NOT NULL,

subjectteaching\_id INTEGER REFERENCES subjectteaching(id) NOT NULL,

auditorium\_id INTEGER REFERENCES auditorium(id)

);

CREATE TABLE lesson

(

id serial PRIMARY KEY,

subjectteaching\_id INTEGER REFERENCES subjectteaching(id) NOT NULL,

time DATE,

auditorium\_id INTEGER REFERENCES auditorium(id)

);

CREATE TABLE privelege

(

id serial PRIMARY KEY,

name VARCHAR(70) NOT NULL,

discount INTEGER NOT NULL

);

CREATE TABLE hostel

(

id serial PRIMARY KEY,

number INTEGER NOT NULL,

head VARCHAR(50) NOT NULL

);

CREATE TABLE roomtype

(

id serial PRIMARY KEY,

name VARCHAR(15) NOT NULL,

price INTEGER NOT NULL

);

CREATE TABLE room

(

id serial PRIMARY KEY,

number INTEGER NOT NULL,

capacity INTEGER NOT NULL,

-- price INTEGER NOT NULL,

hostel\_id INTEGER REFERENCES hostel(id) NOT NULL,

roomtype\_id INTEGER REFERENCES roomtype(id) NOT NULL

);

CREATE TABLE hostelresident

(

id serial PRIMARY KEY,

abilities TEXT,

student\_id INTEGER REFERENCES student(id) NOT NULL,

room\_id INTEGER REFERENCES room(id) NOT NULL,

previlege\_id INTEGER REFERENCES privelege(id)

);

CREATE TABLE cheque

(

id serial PRIMARY KEY,

payment\_date DATE NOT NULL,

sum INTEGER NOT NULL,

start\_date DATE NOT NULL,

end\_date DATE NOT NULL,

hostelresident\_id INTEGER REFERENCES hostelresident(id) NOT NULL

);

CREATE TABLE inventory

(

id serial PRIMARY KEY,

name VARCHAR(70) NOT NULL,

count INTEGER NOT NULL

);

CREATE TABLE doinventory

(

id serial PRIMARY KEY,

inventory\_date DATE,

inventory\_id INTEGER REFERENCES inventory(id) NOT NULL,

hostelresident\_id INTEGER REFERENCES hostelresident(id) NOT NULL

);

ДОДАТОК Б

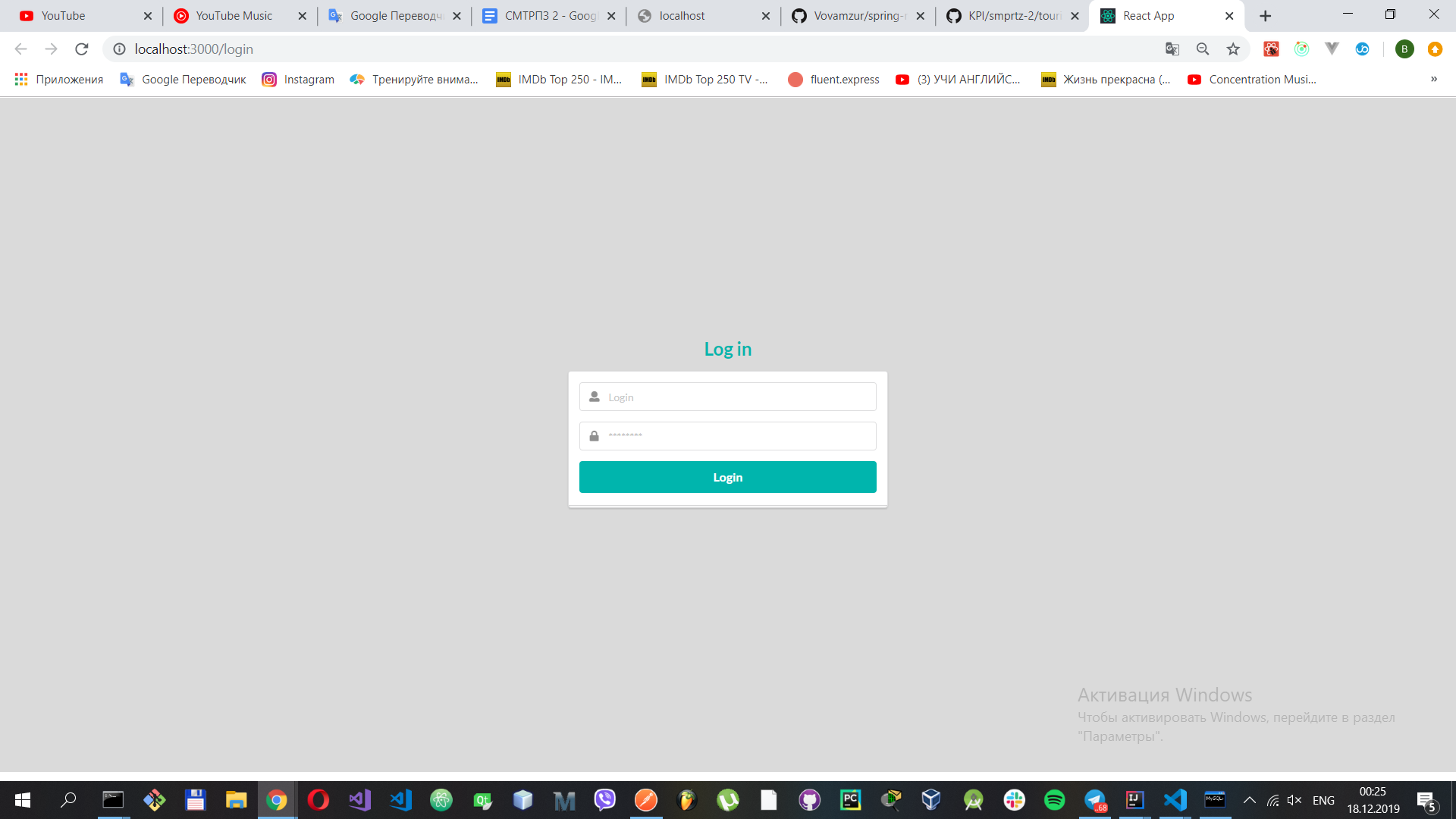
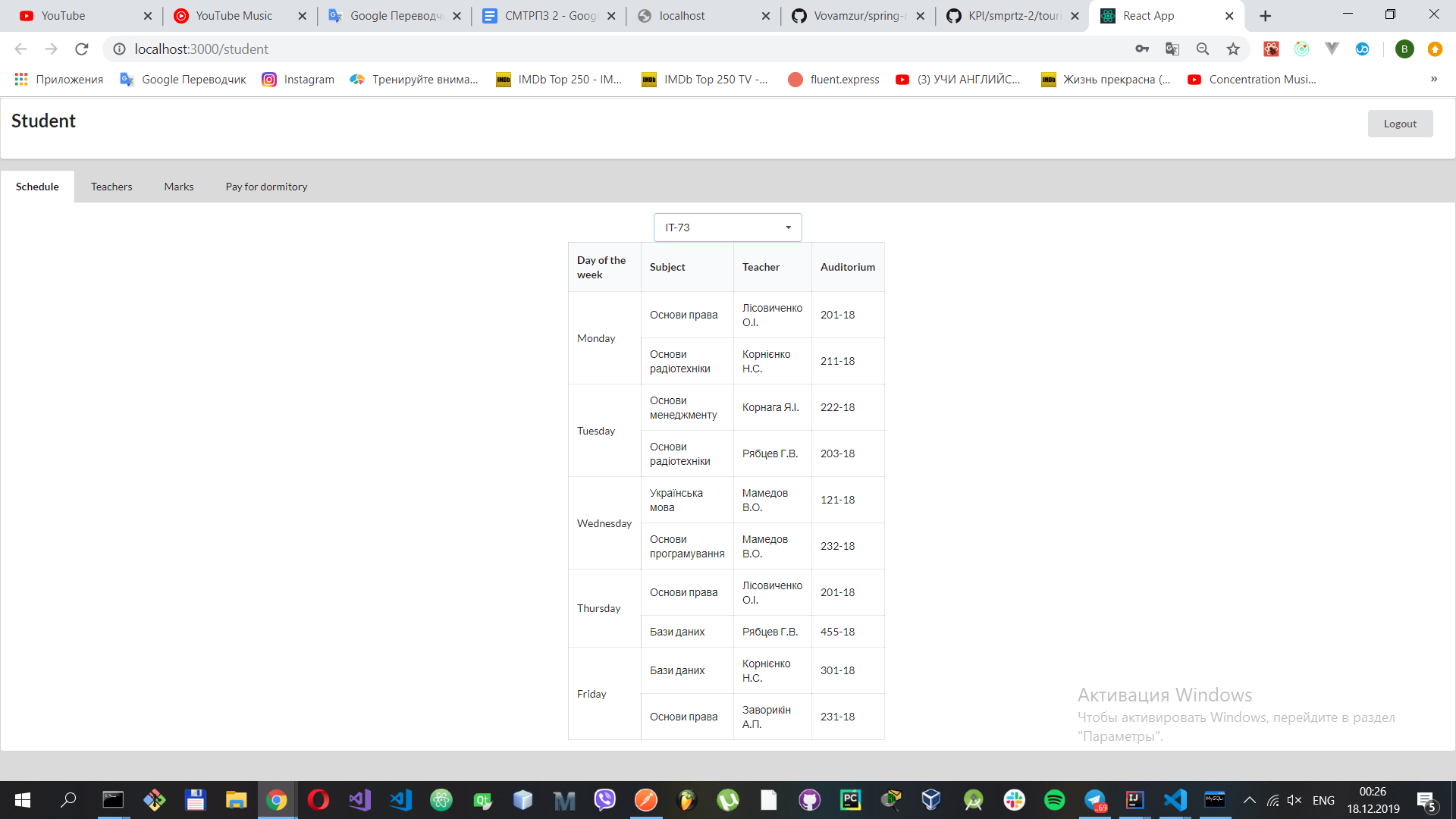


Рис.Б.1 - Логін-пейдж

Рис Б.2 - Головна сторінка студента

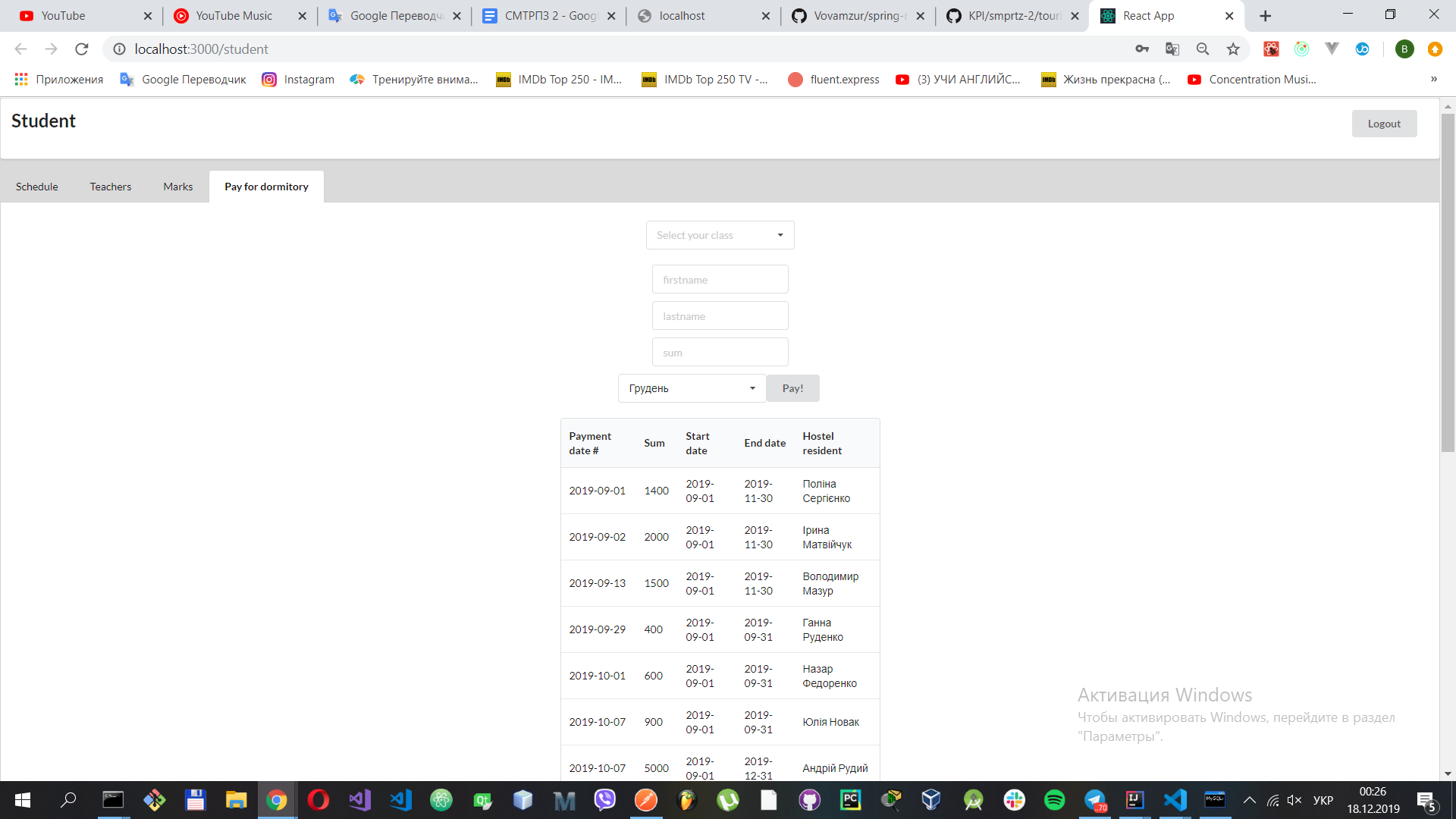


Рис Б.3 - Сторінка оплати проживання у гуртожитку

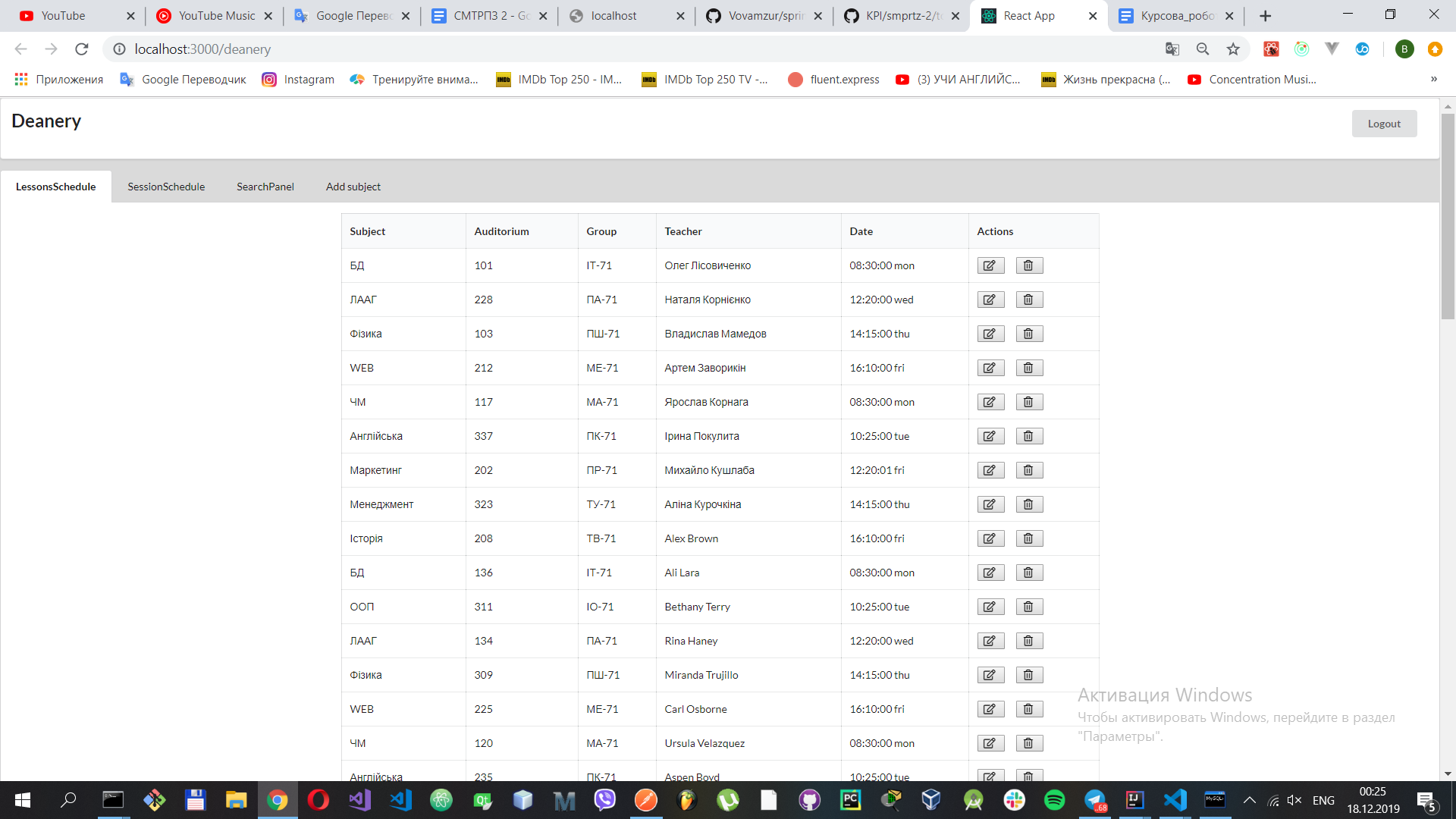


Рис. Б.4 - Головна сторінка працівника деканату

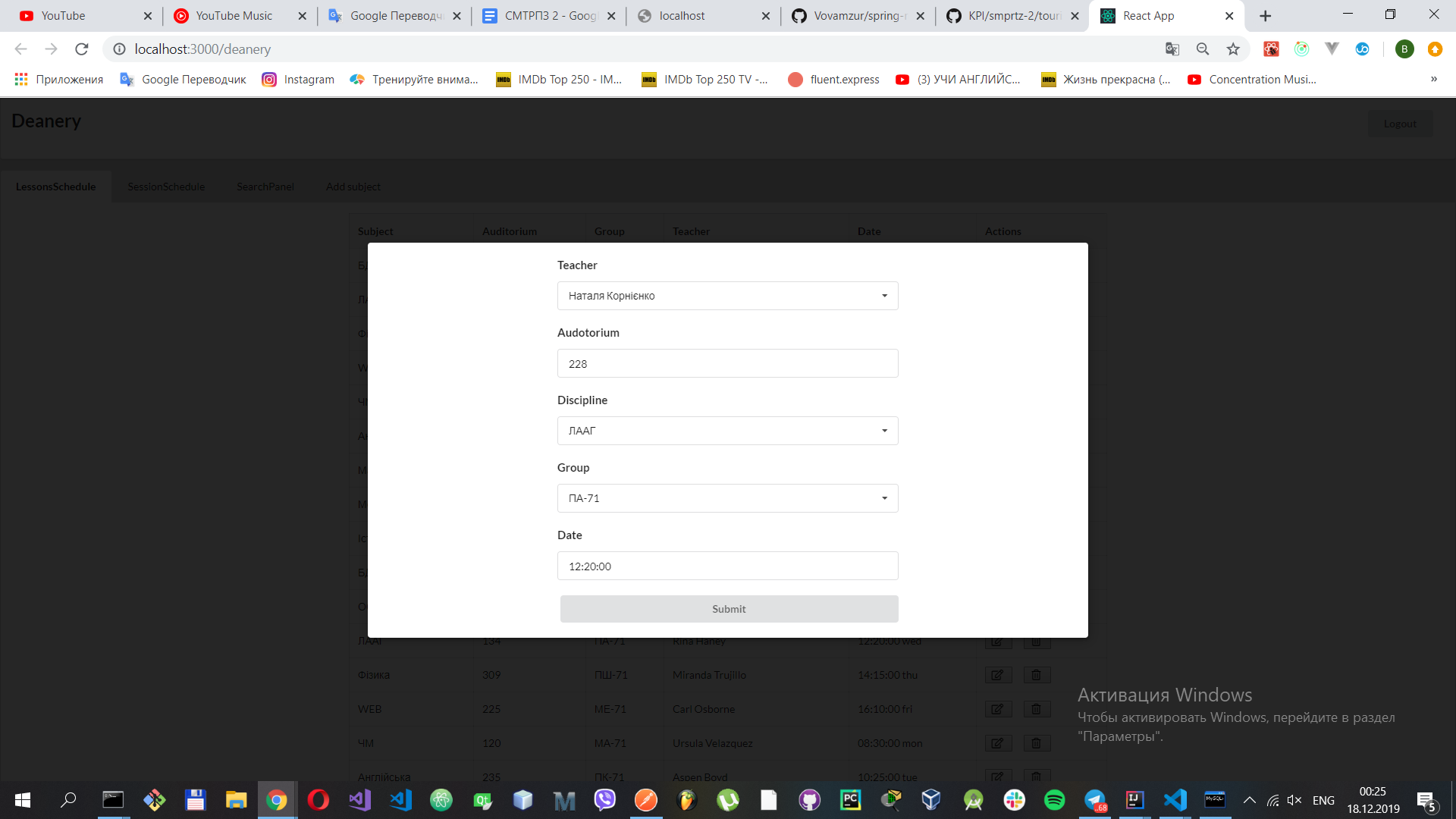


Рис. Б.5 - Виринаюче вікно для редагування розкладу

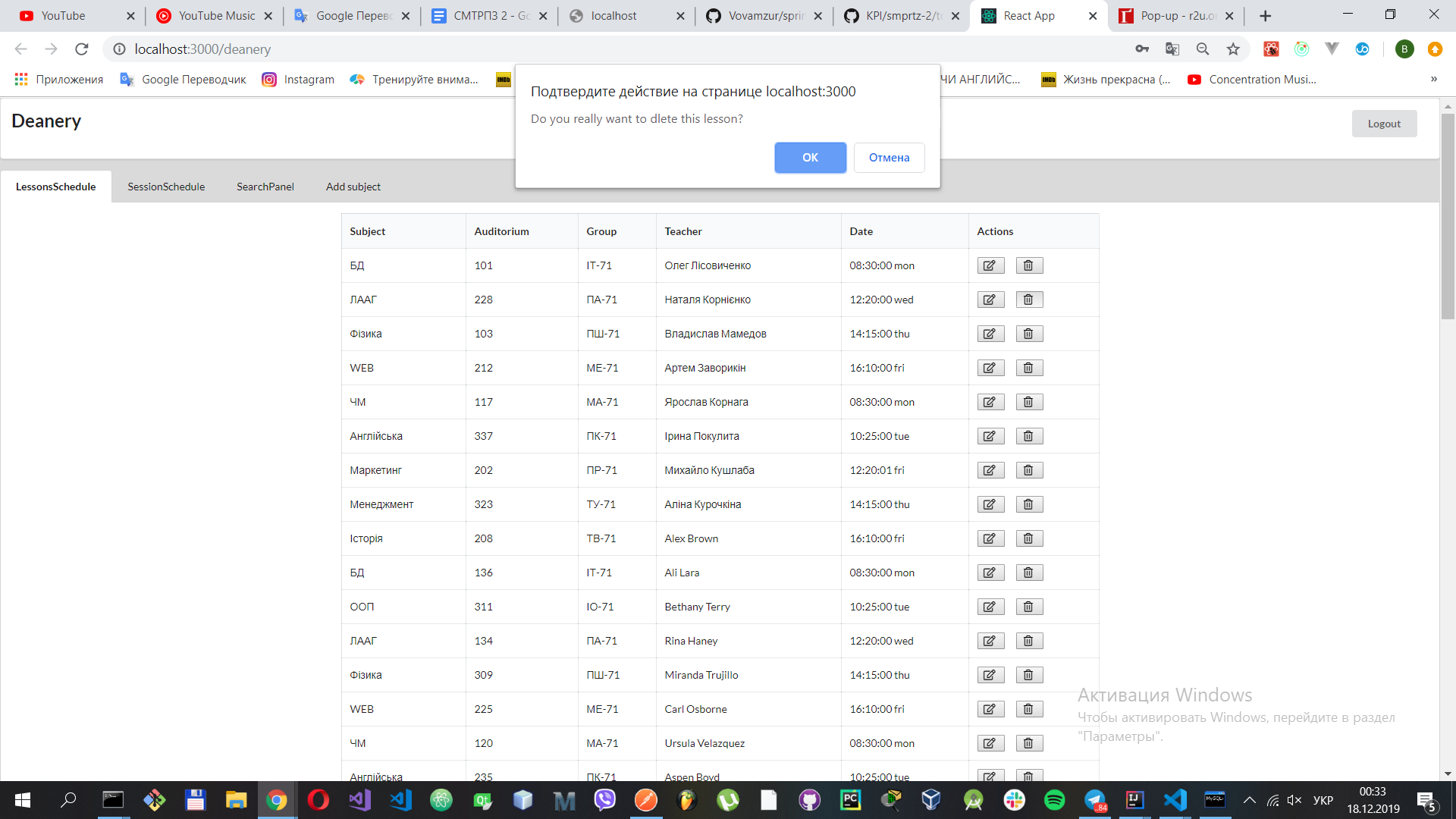


Рис. Б.6 - Приклад видалення певного предмета з розкладу

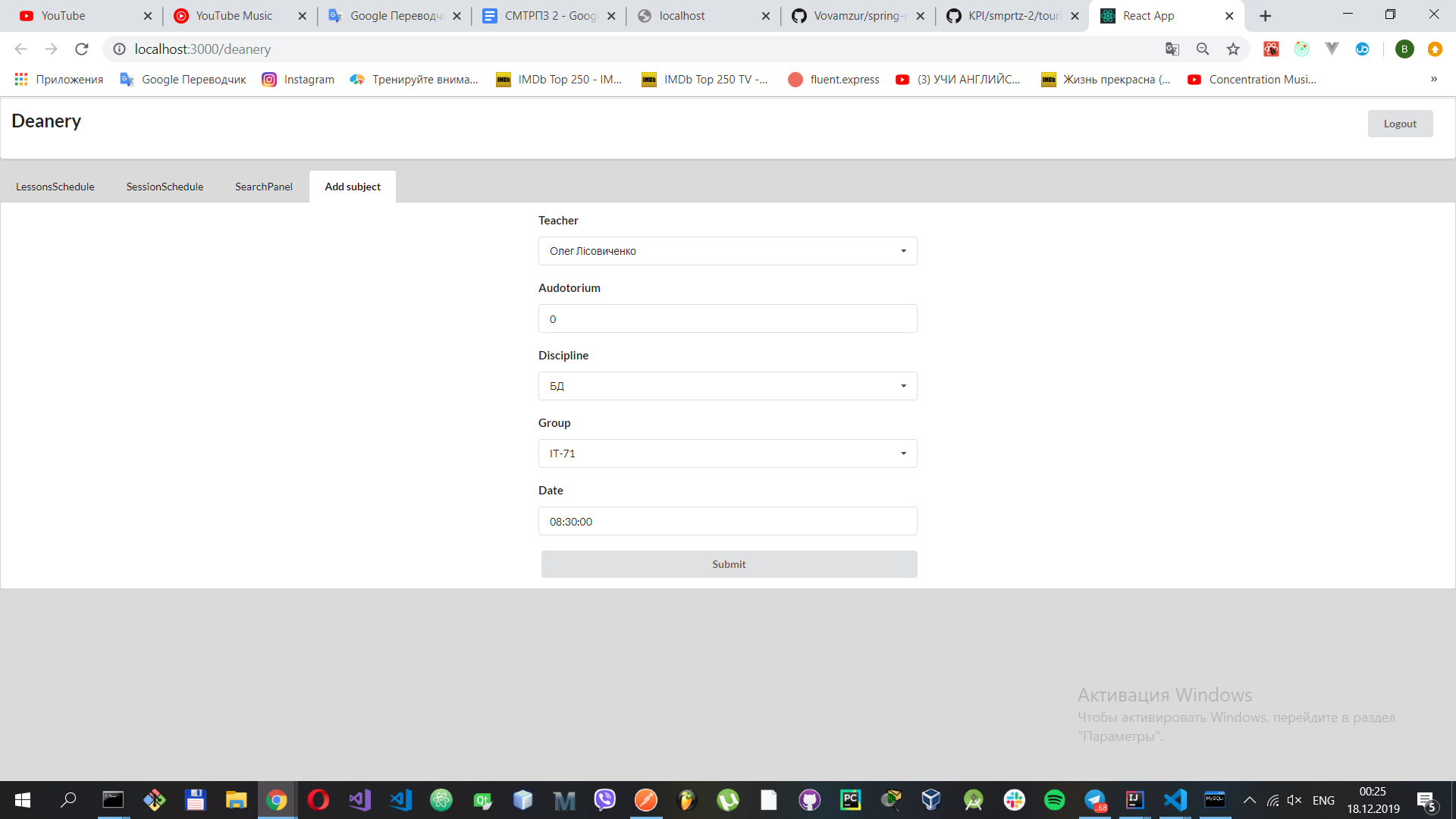


Рис. Б.7 - Сторінка додавання нових елементів в розкладі

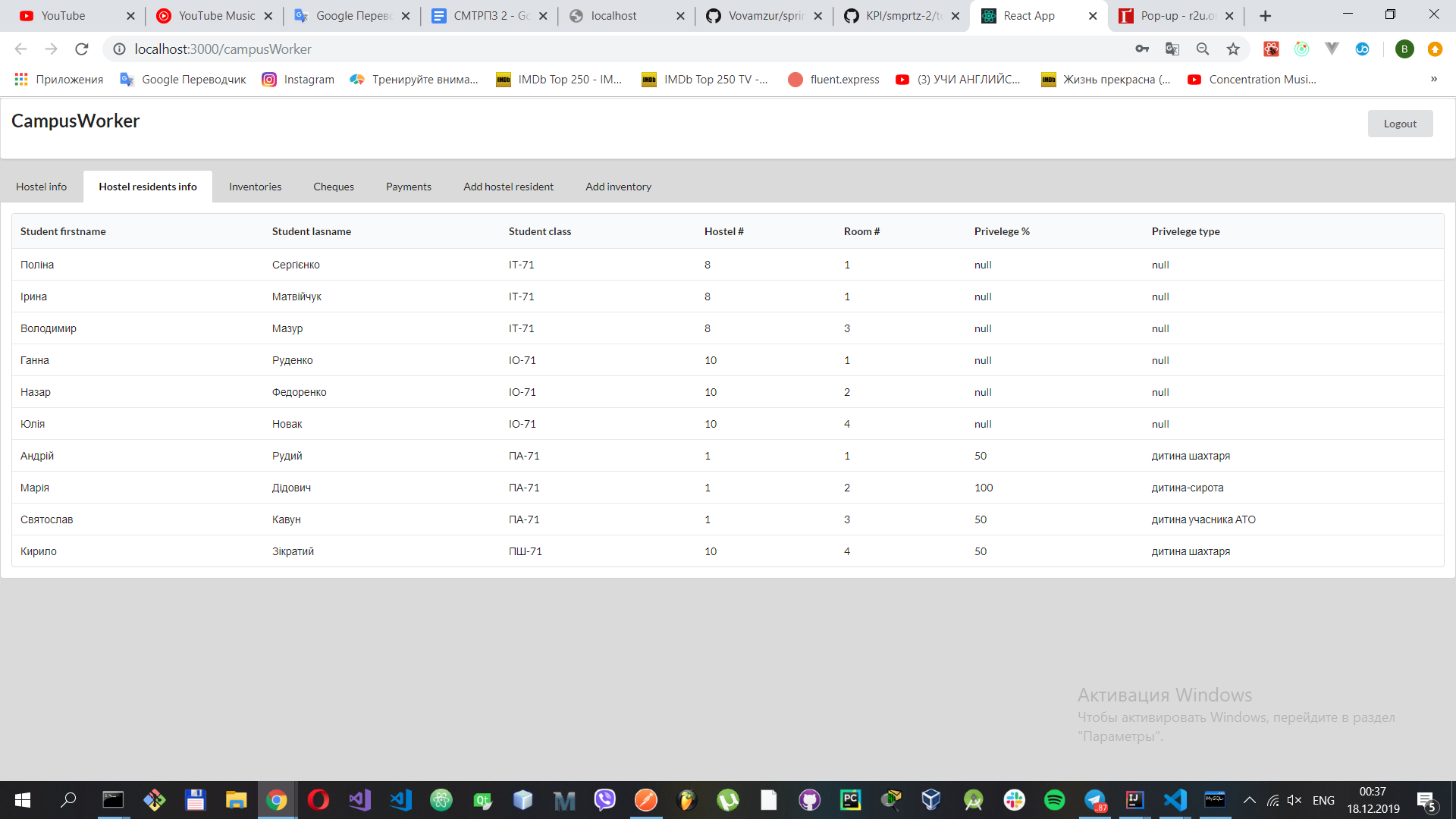


Рис. Б.8 - Головна сторінка працівника кампусу

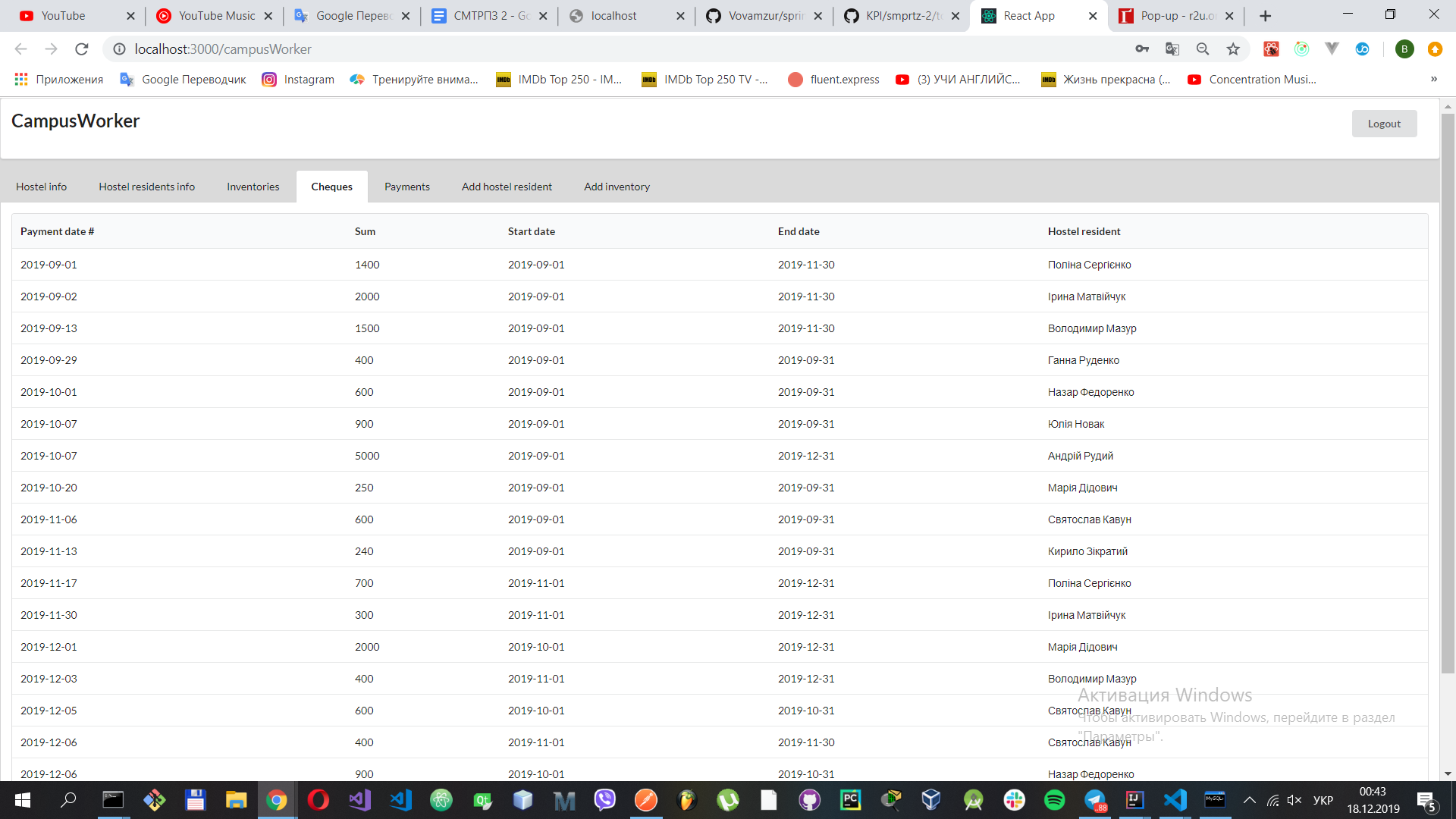


Рис. Б.9 - Сторінка перегляду сплати студентами проживання в гуртожитку

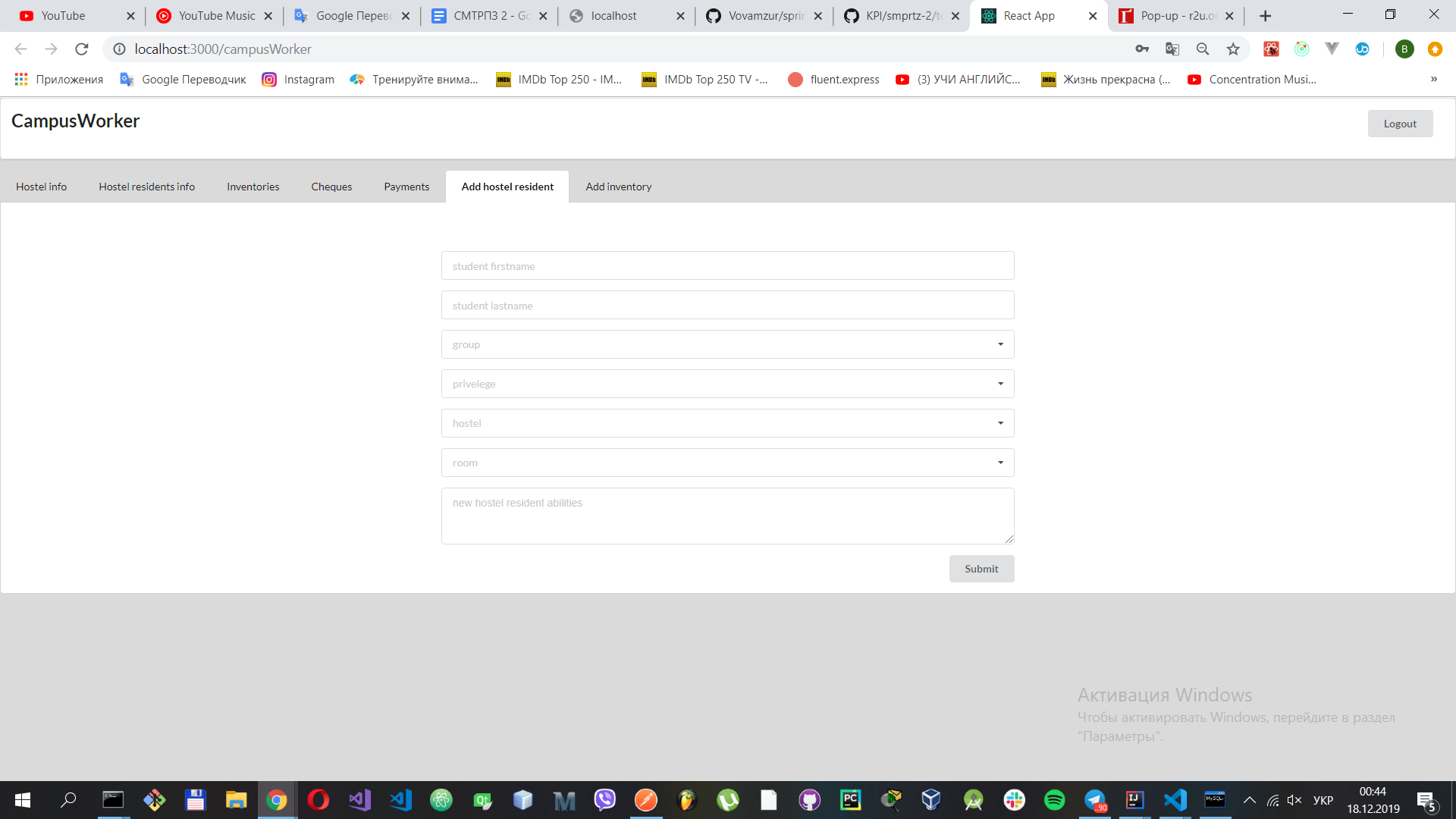


Рис. Б.10 - Сторінка-форма для додавання нових жителів гуртожитка

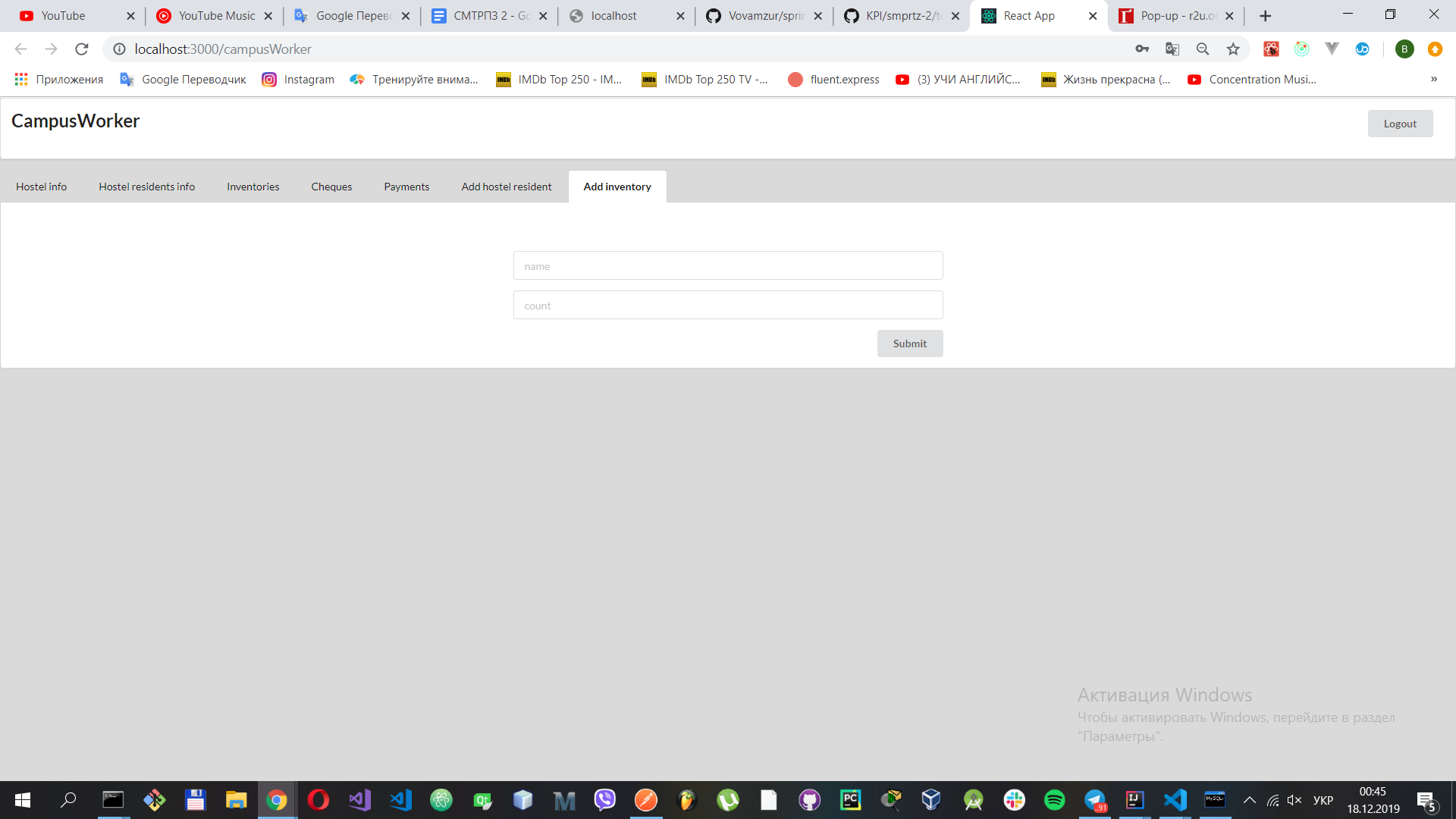


Рис. Б.11 - Сторінка-форма для додавання інвентаря