**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет імені Петра Могили**

**Факультет комп’ютерних наук**

**Кафедра <Інтелектуальних інформаційних систем>**

**ЗВІТ**

*з лабораторної роботи №  1-2*

**Варіант №21**

Дисципліна "**Java EE**"

Спеціальність: 122

122-ЛР.ПЗ.01-1810121

***Cтудент***

*\_\_\_\_\_\_\_\_\_\_\_\_\_ М.О. Салютiн*

*(підпис)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

***Викладач*** *\_\_\_\_\_\_\_\_\_\_\_\_\_К.О.Антiпова*

*(підпис)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**Миколаїв – 2020**

## **Лабораторна робота №1-2**

**Розробка та використання сервлетів при створенні WEB-додатків,**

**розробка графічного web-інтерфейсу з використанням JSF 2.2**

Завдання:

1. У відповідності зі своїм варіантом розробити графічний користувальницький web-інтерфейс, що дозволяє виконувати операції додавання, редагування та видалення даних.
2. Забезпечити можливість пошуку не менше, ніж за 3 критеріями з виведенням на екран результатів пошуку.
3. Функціональність розділити на кілька сервлетів, продемонструвавши при цьому можливості forwarding між Сервлетами.
4. Продемонструвати використання різних видів компонентів CDI.
5. Використовуючи механізми Cookies і Session сервлетів організувати контроль доступу до компонентів (не менше 2-х рівнів доступу, наприклад адміністратор та користувач).

Поліклініка. Таблиці: пацієнт, лікар, прийом.

Прийом пацієнтів проводиться лікарями різних спеціальностей. При оформленні прийому треба вказати інформацію про пацієнта, про лікаря, який консультує пацієнта, про дату прийому, скарги, діагноз.

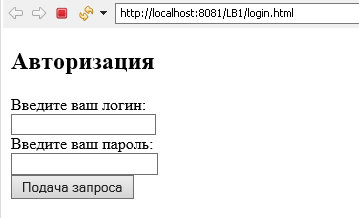
При запуску сайту перед нами з'являється вміст файлу login.html, на якій ми вводимо логін і пароль. Перевіряється все в файлі verification.jsp, після чого нас переводить на сторінку first\_page.html.

Рис.1 – login.html

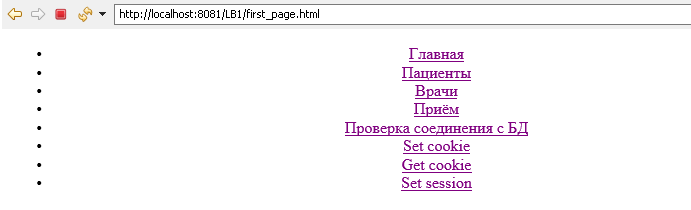


Рис.2 – first\_page.html

На цій сторінці (first\_page.html) нас чекає навігаційне меню, яке дозволяє переміщатися по сторінках (для управління БД), в посиланнях html-тегу «a» зберігаються посилання на сервлети.

<div class=*"navcont"* align=*"center"*>

<div class=*"nav"*>

<ul>

<li><a href=*"first\_page.html"*>Главная</a></li>

<li><a href=*"pacient"*>Пациенты</a></li>

<li><a href=*"doctor"*>Врачи</a></li>

<li><a href=*"priyom"*>Приём</a></li>

<li><a href=*"hospitalbd"*>Проверка соединения с БД</a></li>

<li><a href=*"setcookie"*>Set cookie</a></li>

<li><a href=*"getcookie"*>Get cookie</a></li>

<li><a href=*"setsession"*>Set session</a></li>

</ul>

</div>

</div>

Зараз будуть скріншоти кожної з посилань:

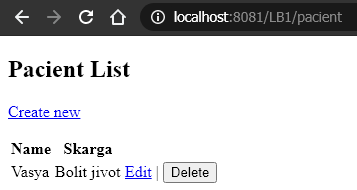


Рис.3 – сервлет PatientShowServlet.java

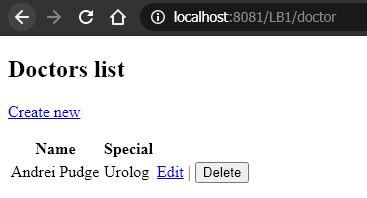


Рис.4 – сервлет DoctorShowServlet.java



Рис.5 – сервлет PriyomShowServlet.java

Метод forward () класу RequestDispatcher дозволяє перенаправити запит з сервлета на інший сервлет, html-сторінку або сторінку jsp, що ми і використовували для forwad`інга між Сервлетами.

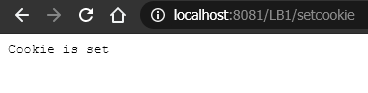


Рис.6 – сервлет SetCookieServlet.java

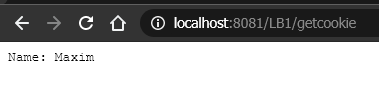


Рис.7– сервлет GetCookieServlet.java

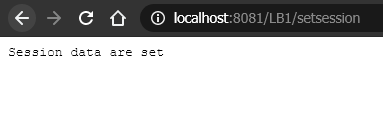


Рис.8– сервлет SetSessionServlet.java, ставиться сессiя



Рис.9 – оновлена сторiнка пiсля поставленої сесciї

Завдання:

1. Розробити форму вводу даних (CRUD-операції) за своїм варіантом, використовуючи вбудовані jsf-компоненти та композитні компоненти.
2. Розділити прототип програми для моделі MVC згідно з варіантом на шари – контролери, робота з бд, інтерфейс.
3. За допомогою JDBC (MySQL, PostgreSQL тощо) створити CRUD функції для роботи з базою даних згідно з варіантом.

Для виконання другої лабораторної роботи було використано програмне забезпечення OpenServer, а в ньому використовувався phpMyAdmin для створення бази даних.

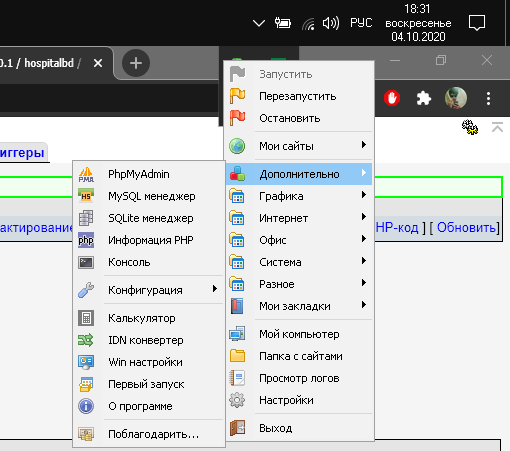


Рис.10 – запуск phpMyAdmin.

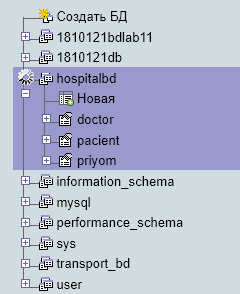


Рис.11 – бази даних, які у мене є, для роботи знадобиться hospitalbd.

Вона складається з трьох таблиць:

1.Доктор (поля: id, name, special).

2.Паціент (поля: id, name, skarga).

3. Прийом(поля: id, skarga, diagnos, datapr).

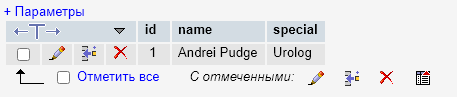


Рис.12 – таблиця doctor.

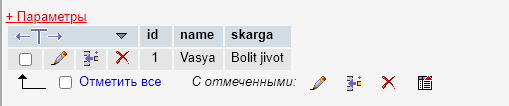


Рис.13 – таблиця pacient.

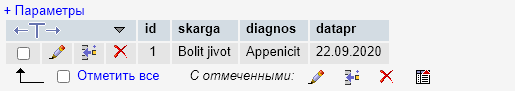


Рис.14 – таблиця priyom.

CRUD - акронім, що позначає чотири базові функції, які використовуються при роботі з базами даних: створення (англ. Create), читання (read), модифікація (update), видалення (delete).

Всі ці функції реалізовані в наших сервлетах і певних класах java, які реалізують роботу з полями бази даних.



Рис.15 - це стандартний набір класів Java Beans. Він представляє дані, з якими ми будемо працювати в додатку, і відповідає структурі даних в таблиці hospitalbd в базі даних.

Перш за все клас визначає загальні установки з'єднання. Мінлива url задає адресу для підключення до бд MySQL, username представляє логін, а password - пароль від сервера MySQL.

Клас визначає всі основні операції з базою даних. Метод select () отримує всі дані з таблиці hospitalbd і створює з них список об'єктів для таблиць Doctor, Pacient, Priyom.

1. Метод selectOne () отримує один елемент з таблиці по id. Якщо об'єкт в бд не знайдений, то повертається null.
2. Метод insert () додає в таблицю один об'єкт Product і повертає кількість доданих рядків.
3. Метод update () оновлює в таблиці один об'єкт Product і повертає кількість оновлених рядків.
4. Метод delete () видаляє один об'єкт Product і повертає кількість вилучених рядків.

Для спрощення роботи з базою даних також визначимо в тому ж пакеті новi класи – DoctorBD, PacientBD, PriyomBD.

Функції роботи з базою даних:

1.Додавання

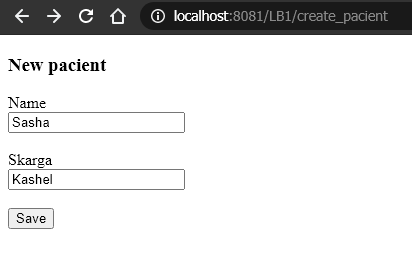


Рис.16 – сервлет CreatePacientServlet.java

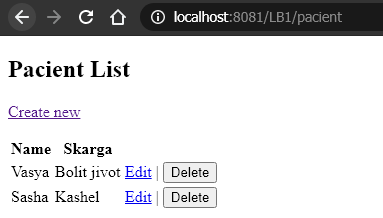


Рис.17 – результат роботи додавання



Рис.18 – результат роботи додавання для перевiрки, додалося чив воно в phpMyAdmin.

2. Редагування

Відредагуємо другого пацієнта, помінявши йому ім'я на Данило.(по id)

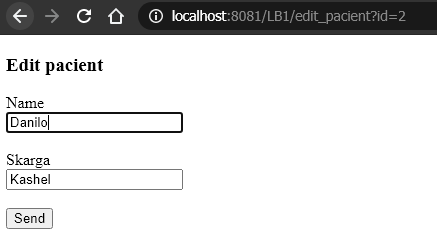


Рис.19 - сервлет EditPacientServlet.java

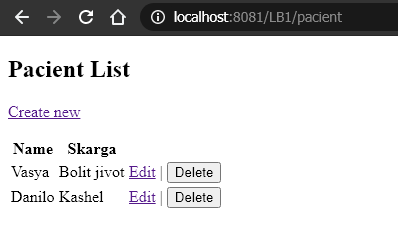


Рис.20 – результат роботи.

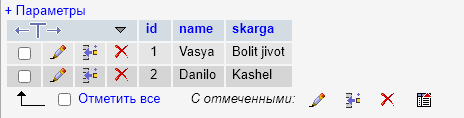


Рис.21 - результат роботи додавання для перевiрки, редагувалось чи воно в phpMyAdmin.

3.Видалення

Видалимо другого пацієнта, натиснувши для цього на кнопку Delete.

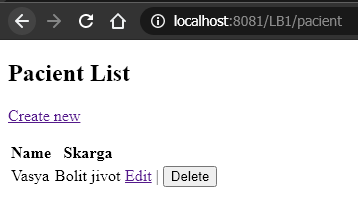


Рис.22 - результат роботи сервлету DeletePacientServlet.java.

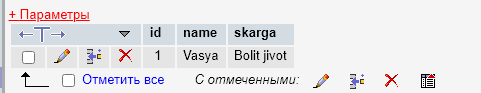


Рис.23 - результат роботи .

Iншi дві таблиці мають такий самий функціонал, все це зроблено завдяки сервлетам і .jsp файлів, які допомагають в процесі виводити інформацію, яку обробляє сервлет. Список всіх сервлетiв, для роботи з базою даних.

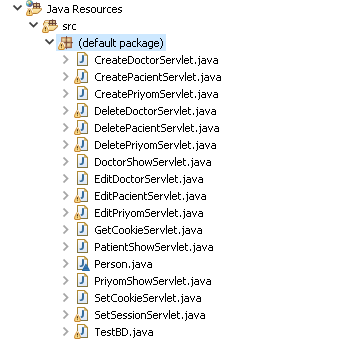


Рис.24 - список всіх сервлетiв, для роботи з базою даних.

Перед роботою з базою даних ми можемо перевірити, стабільне наше підключення, або є воно взагалі. Для цього ми зробили сервлет TestBD.java.

При підключенні йдуть всі перевірки і передаються потрібні змінні для вдалого з'єднання. Якщо ж щось встановлено неправильно: рядок підключення, логін, пароль, що не завантажен драйвер або щось ще, то ми побачимо повідомлення про помилку.

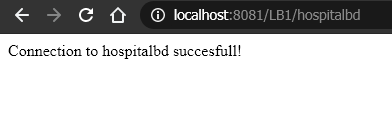


Рис.25 – результат роботи сервлету TestBD.java

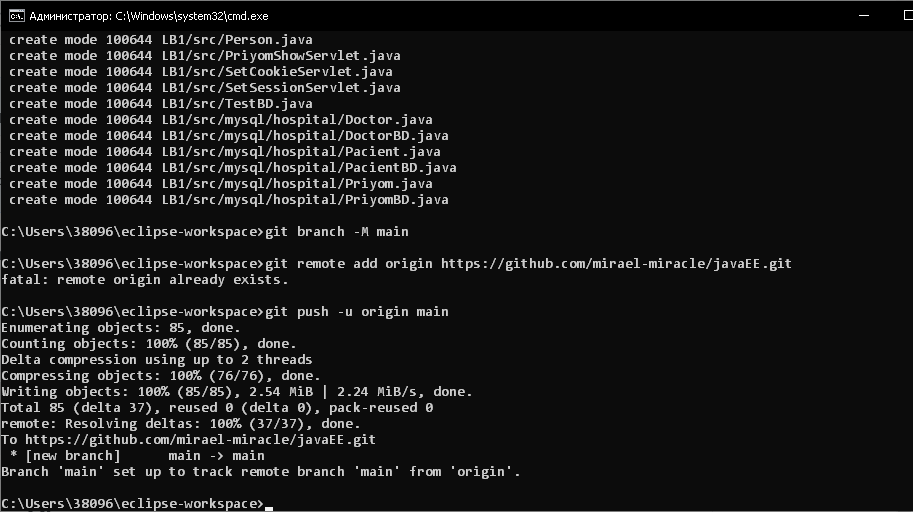


Рис.26 – додавання на github.

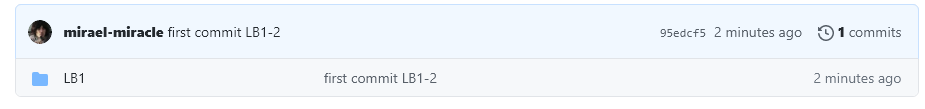


Рис.27 – результат роботи.

Посилання на проект GitHub - <https://github.com/mirael-miracle/javaEE.git>

Висновок: в результаті роботи було вивчено та освоєно - CRUD-операції, розроблений web-інтерфейс з можливістю керувати ним, розділили функціональність на пару сервлетів і змогли використати forwading.