

Курсовий проект з предмету ІПС

Виконали: Арзамасцев О. В., Дорошенко Л. Ю., ІО-52.

Тема: Веб-сайт: Ресторан.

Короткий опис проекту.

Даний проект виконаний у вигляді клієнт-серверної програми мовою Java 10 з використанням технології Java EE Servlets+JSP+EL+JSTL, в якості веб-контейнера використаний Tomcat 9. Для роботи з базою даних використовується JDBC API+MySQL. Фреймворк для тестування – Junit 5, в якості сторонніх бібліотек для тестування використані AssertJ та Mockito.

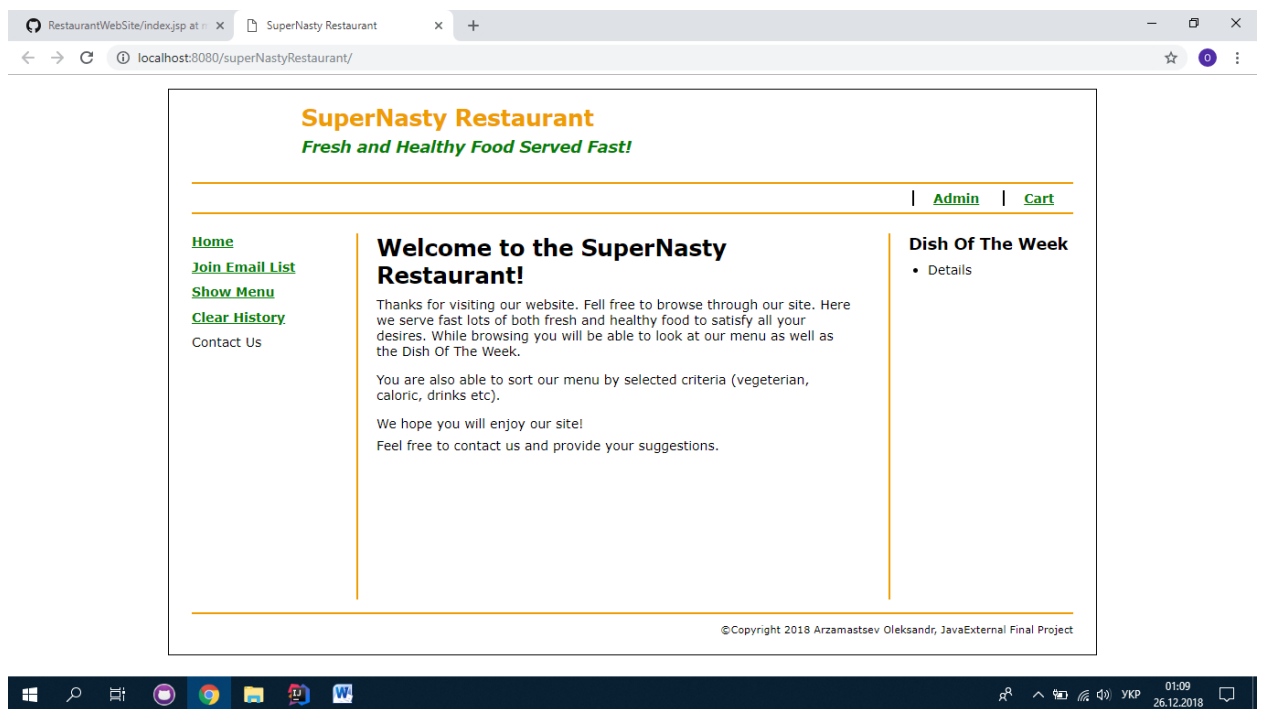


Рис. 1. Початкова сторінка сайту.

Tomcat впізнає домашню сторінку із списку файлів директорії webapp за рахунок того, що у дескрипторі розгортання (deployment descriptor, WEB-INF/web.xml) в розділі welcome-file-list вказано сторінку з розширенням .jsp, а саме index.jsp, яку ми і бачимо на Рис. 1. На цій сторінці є різні посилання-переходи на інші сторінки (Join Email List, Show Menu etc.), натискаючи на дані посилання користувач може перейти в інші частини сайту.



Рис. 2. Меню.

На даному рисунку зображено меню ресторану. Меню завантажується із бази даних, описаній у файлі `restaurant_createDB.sql`. Завантаження меню із бази даних відбувається під час ініціалізації класу `RestaurantContextListener`, що реалізує інтерфейс `ServletContextListener`. Дані, необхідні для подальшої роботи всієї програми також завантажуються в даному класі.

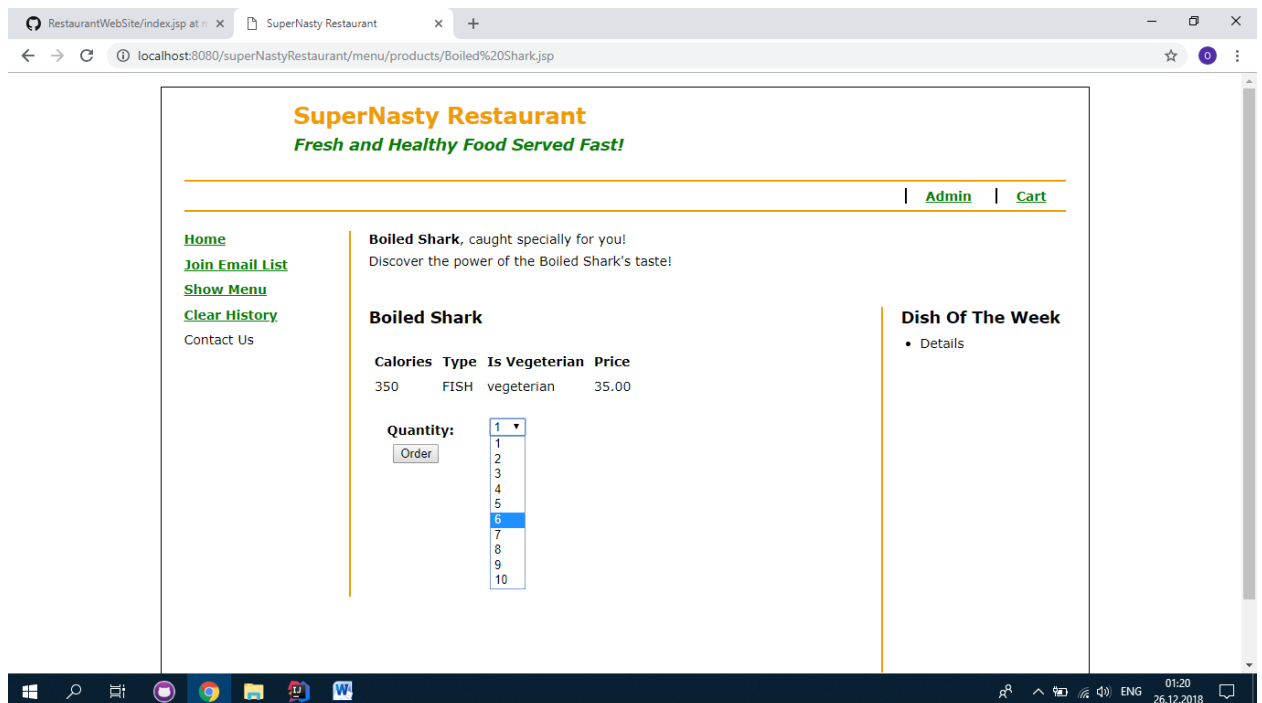


Рис. 3. Приклад замовлення страви із меню.

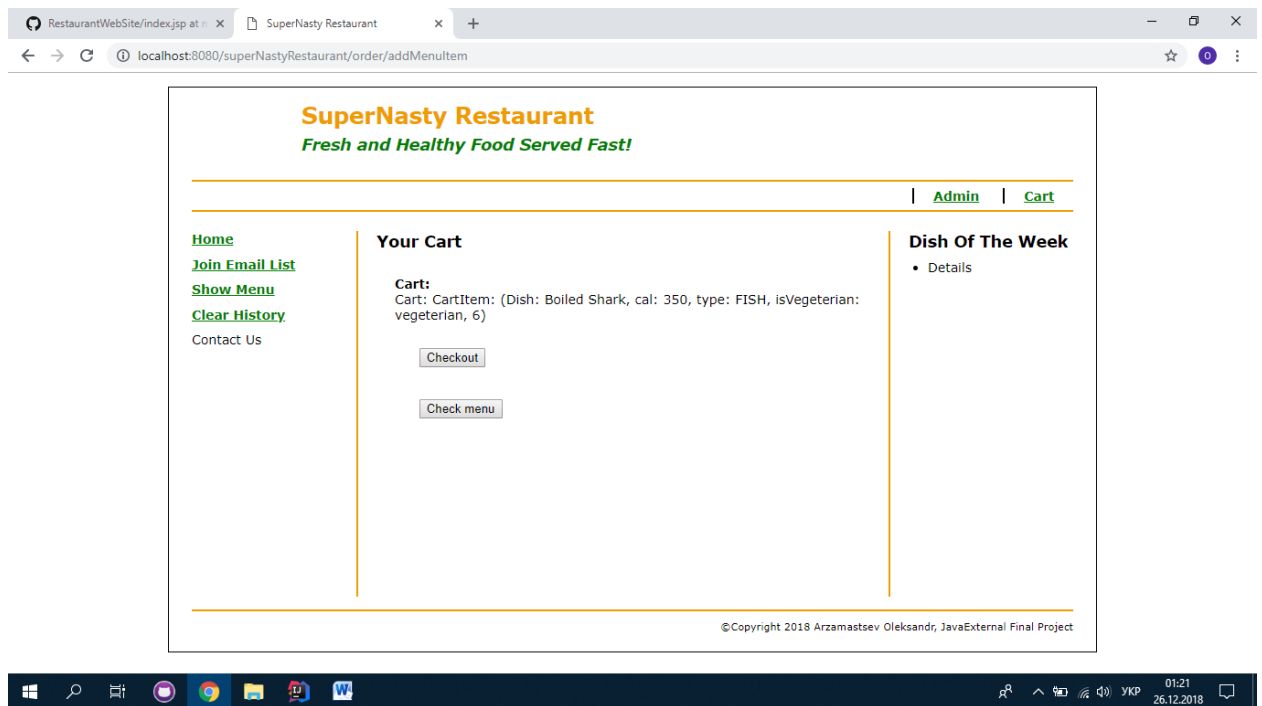


Рис. 4. Корзина користувача

На даній сторінці користувач може або продовжити обирати страви із меню, або ж продовжити оформлювати замовлення.

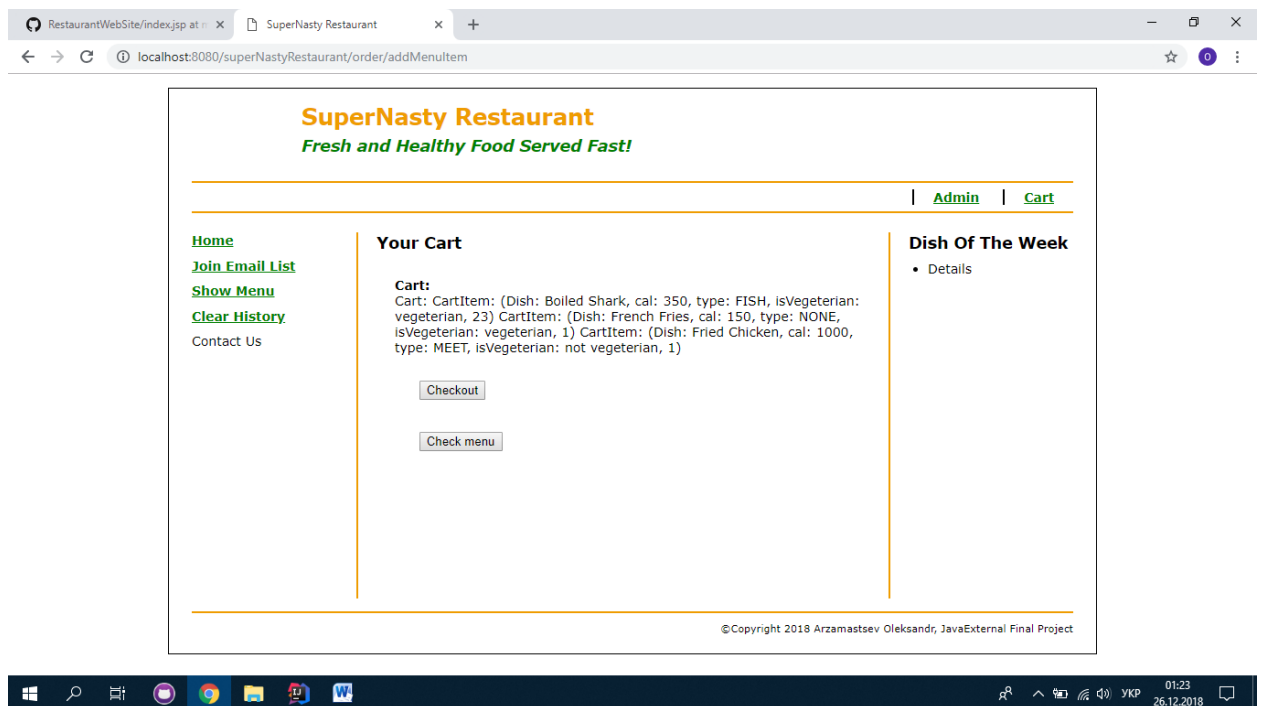


Рис. 5. Користувач продовжив обирати страви із меню.

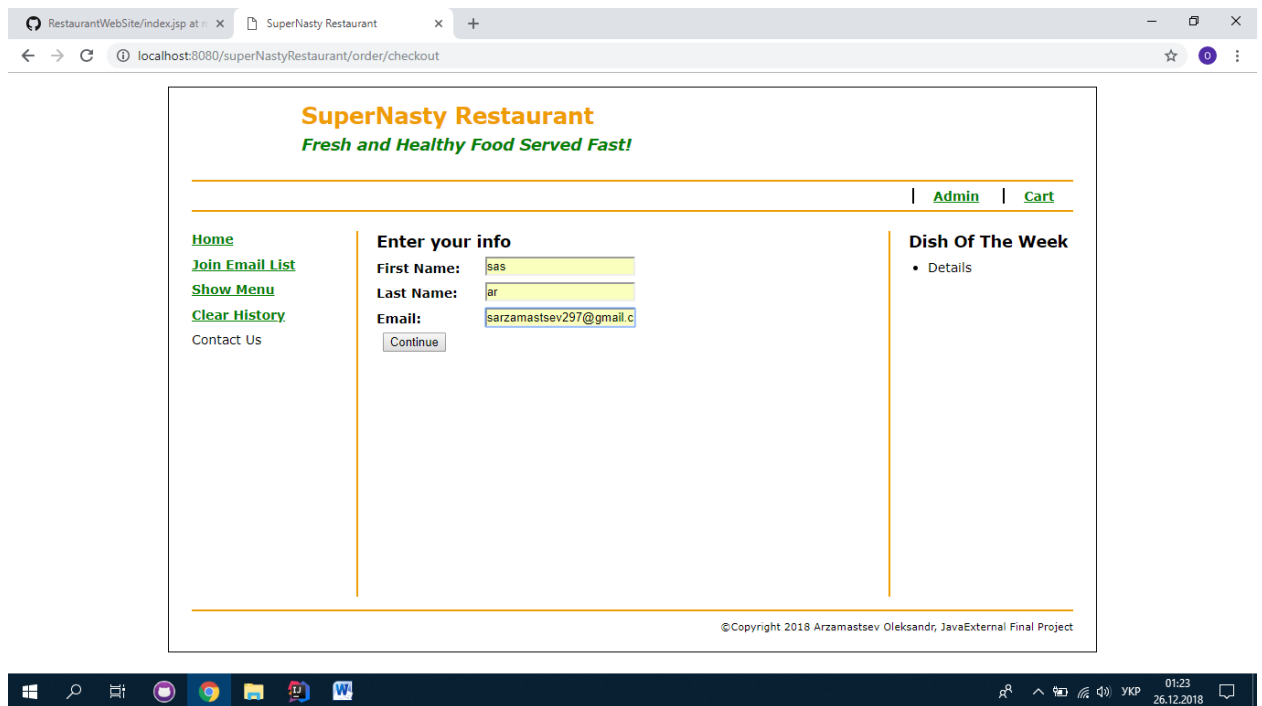


Рис. 6. Користувач продовжує оформлення замовлення.

На даній сторінці користувач вводить своє ім'я, прізвище та адресу електронної пошти. В середині логіка програми побудована таким чином, що ці дані зберігаються за допомогою механізму Sessions+Cookies, що дозволить користувачу в разі випадкових переходів на небажані розділи сайту не втрачати вже заповнену інформацію.

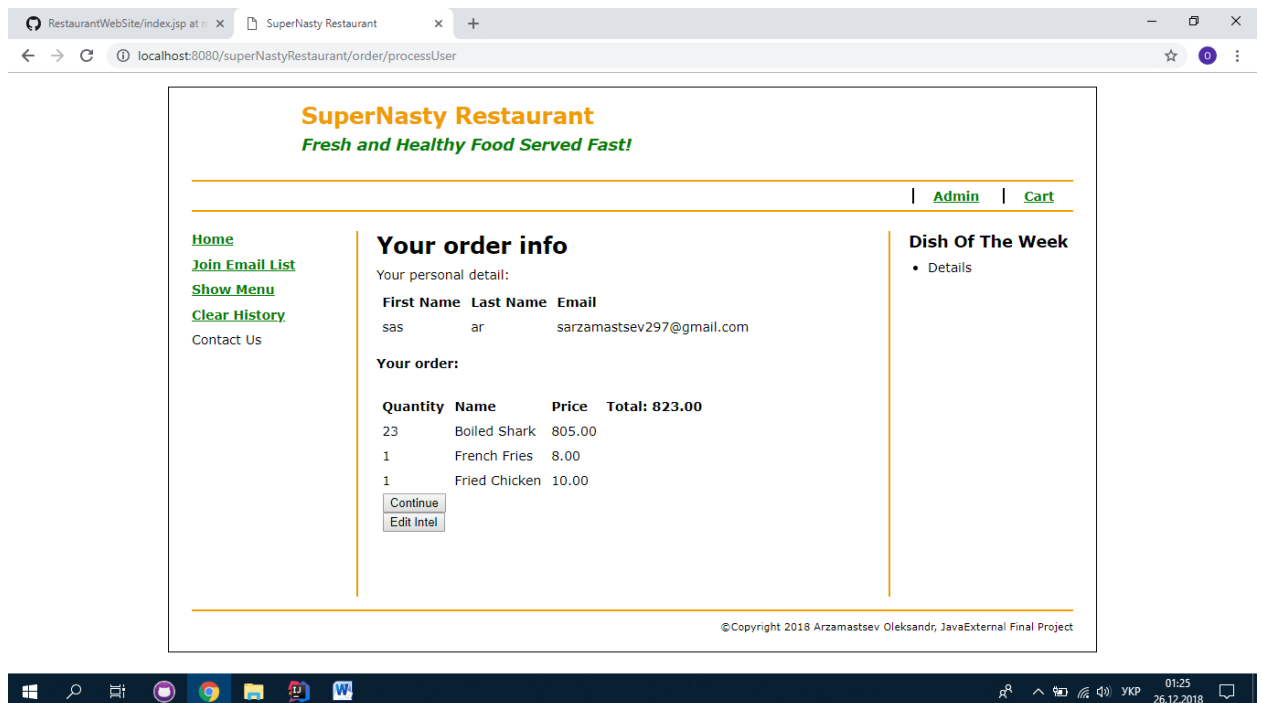


Рис. 7. Деталі замовлення.

На даній сторінці користувач може редагувати персональні дані, або ж продовжити оформлення замовлення.

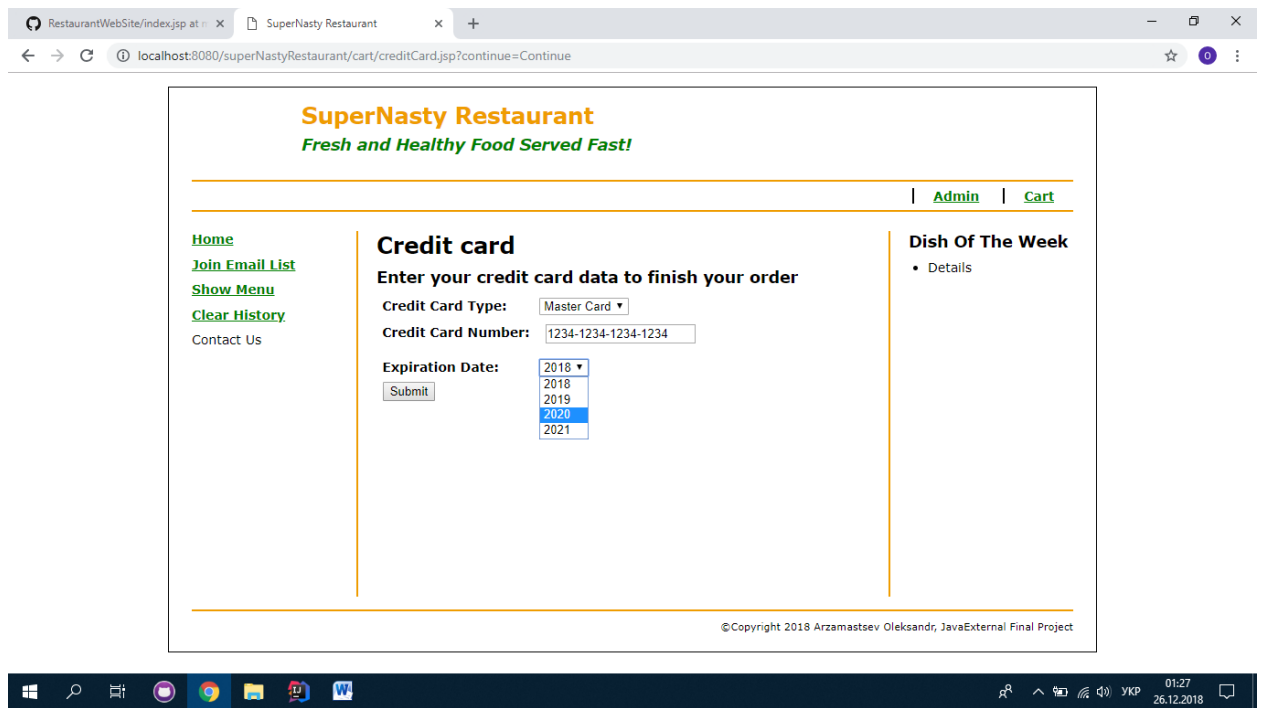


Рис. 8. Дані кредитної картки.

Після заповнення всієї необхідної інформації користувач підтверджує замовлення, натискаючи на кнопку Submit.

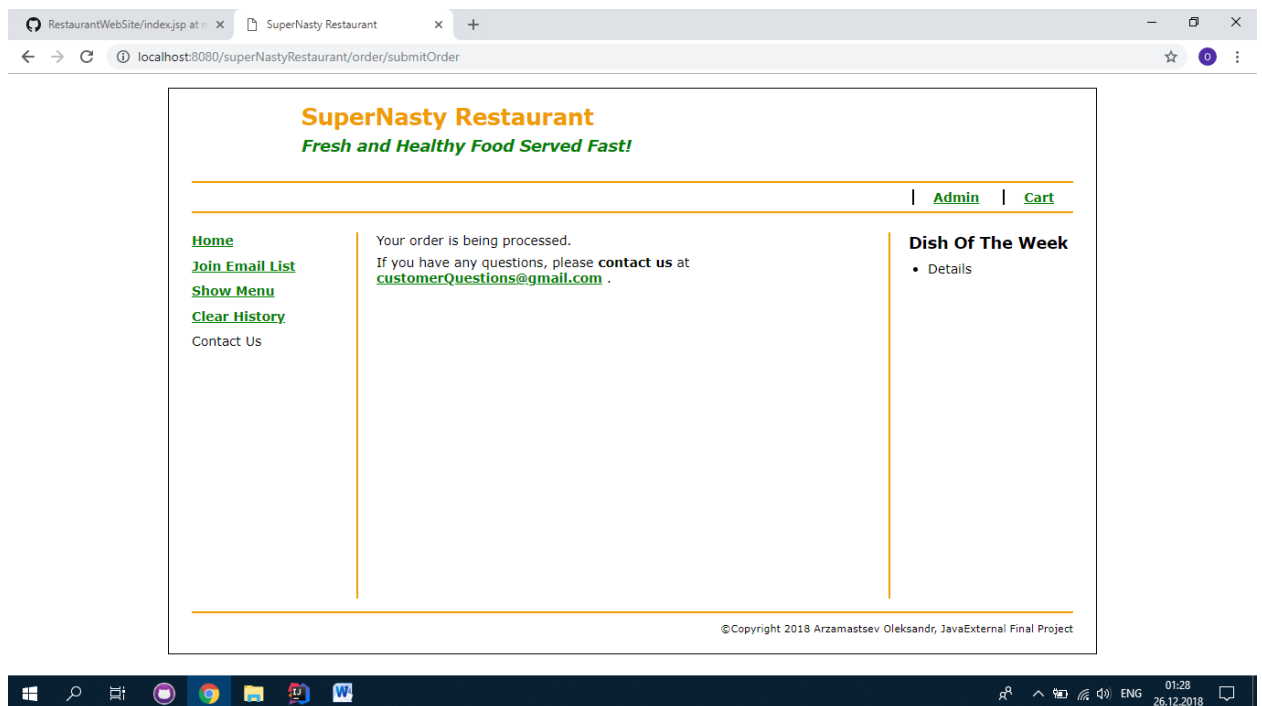


Рис. 9. Успішне оформлення замовлення завершено.

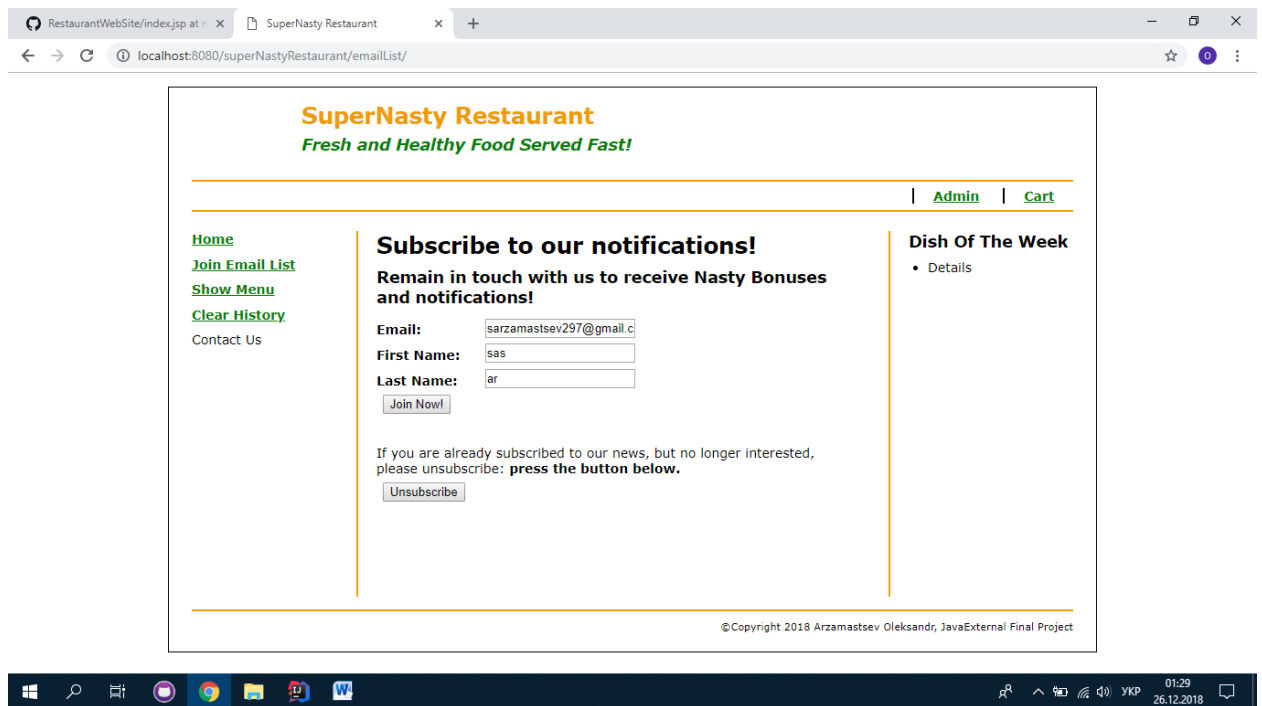


Рис. 10. Вікно підписки для отримання повідомлень.

На даній сторінці користувач може ввести адресу пошти та своє ім'я і таким чином підписатись на новини сайту.

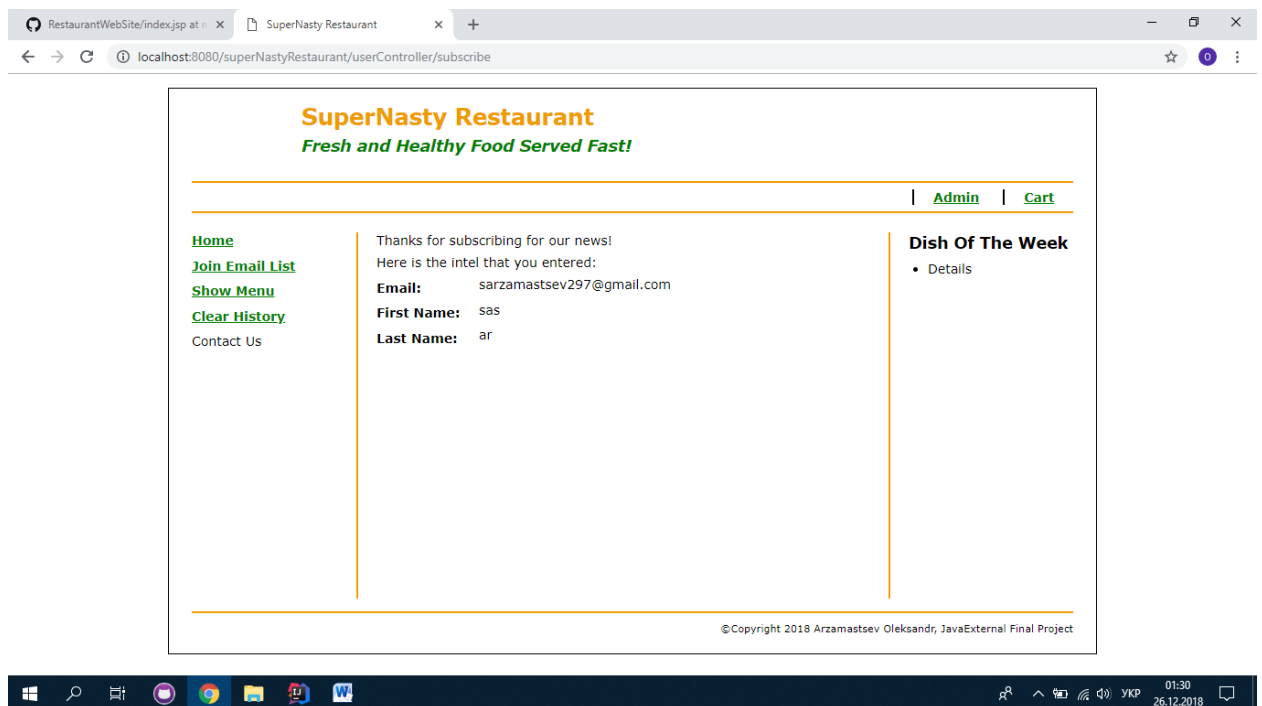


Рис. 11. Успішна підписка на новини.

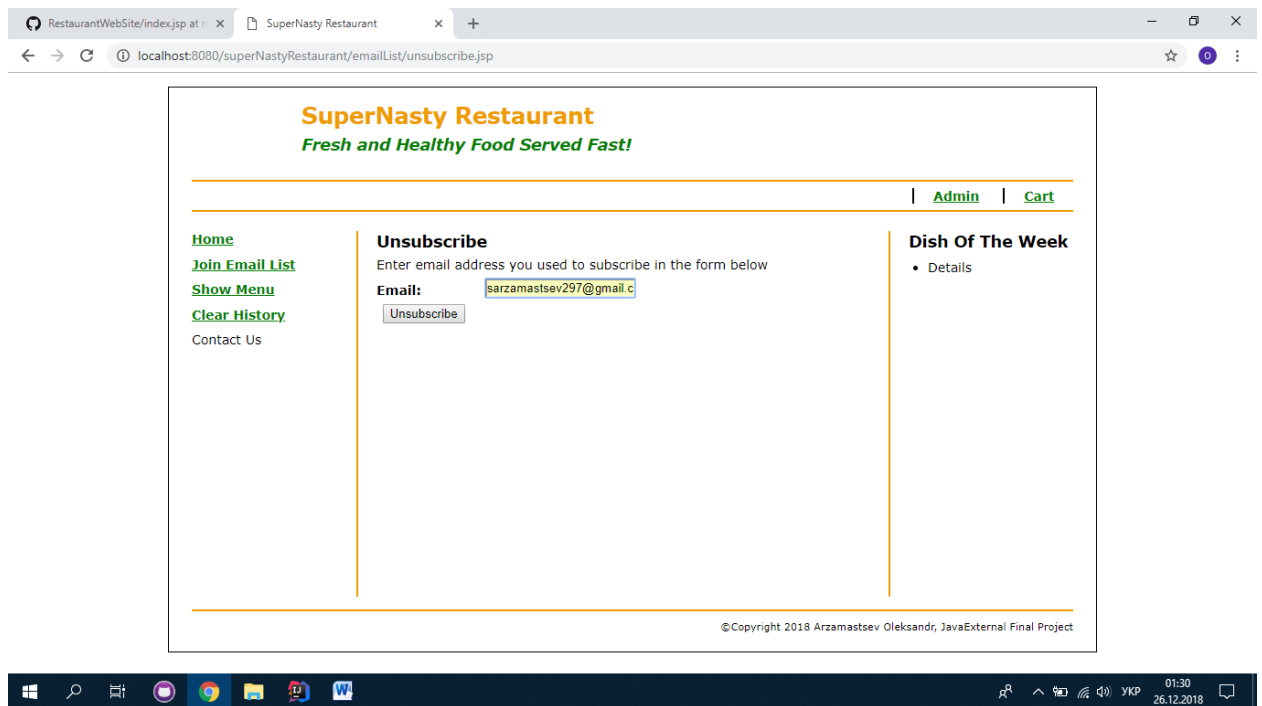


Рис. 12. Сторінка скасування підписки на розсилку новин.

Якщо користувач має попередню підписку, то натиснувши на кнопку Unsubscribe користувач скасовує підписку, вважаючи, що адреса електронної пошти є в базі даних ресторану.

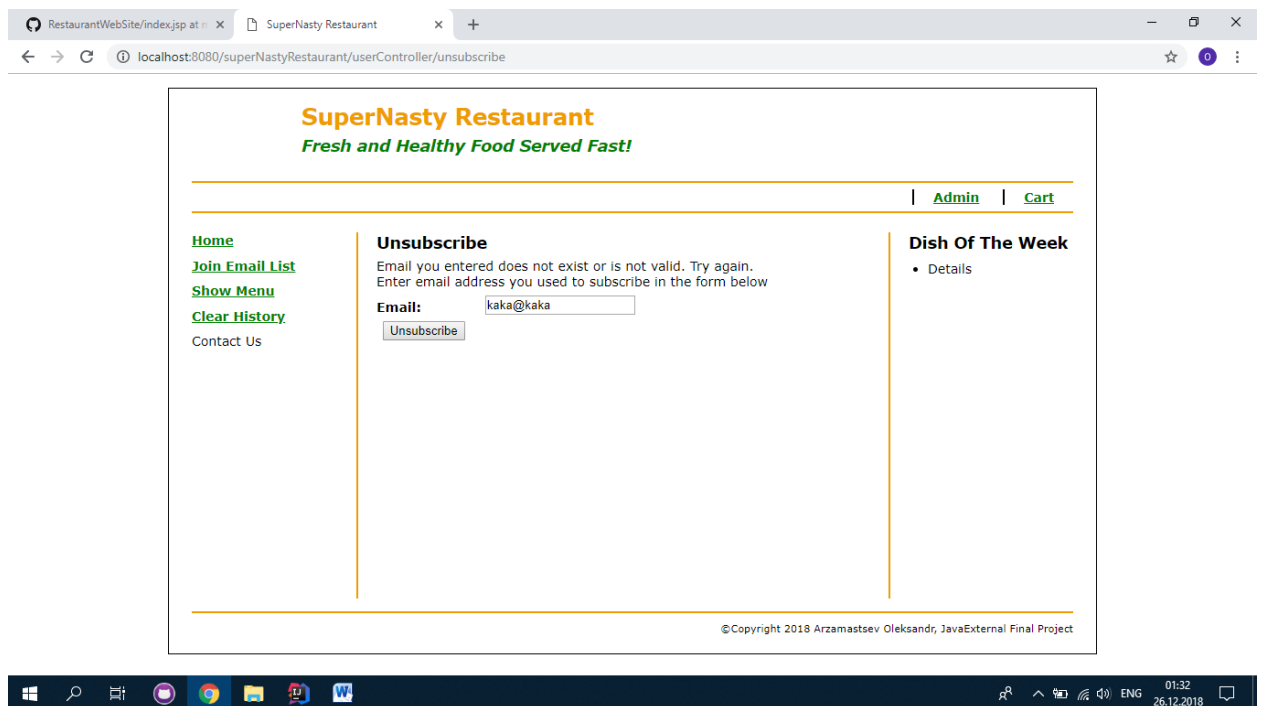


Рис. 13. Неуспішне скасування підписки.

Якщо такої адреси електронної пошти немає в базі даних – отримуємо відповідне повідомлення.

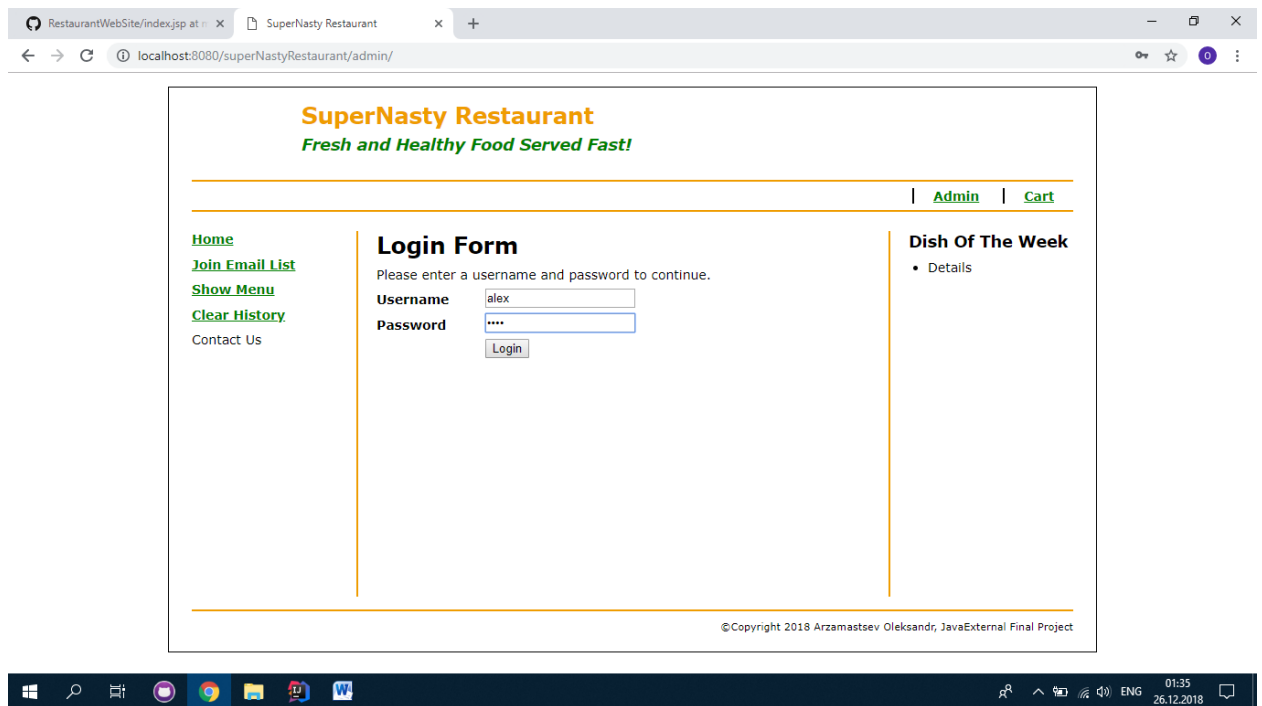


Рис. 14. Авторизація адміністратора.

Для виконання адміністративних функцій на сайті потрібно виконати авторизацію. Авторизація в даному проекті виконана за допомогою використання прийому JDBC-Realm, який зчитує дані логінів та відповідні паролі із бази даних, деталі якої вказані в файлі context.xml.

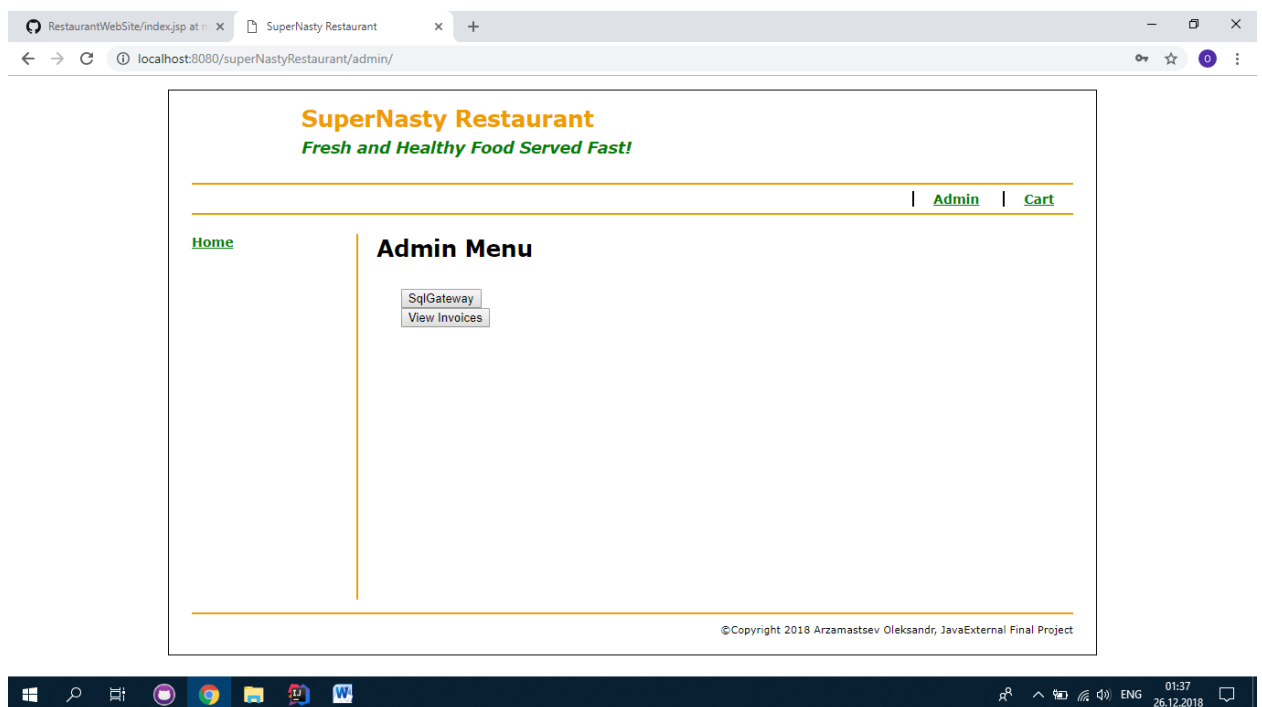


Рис. 15. Сторінка адміністратора.

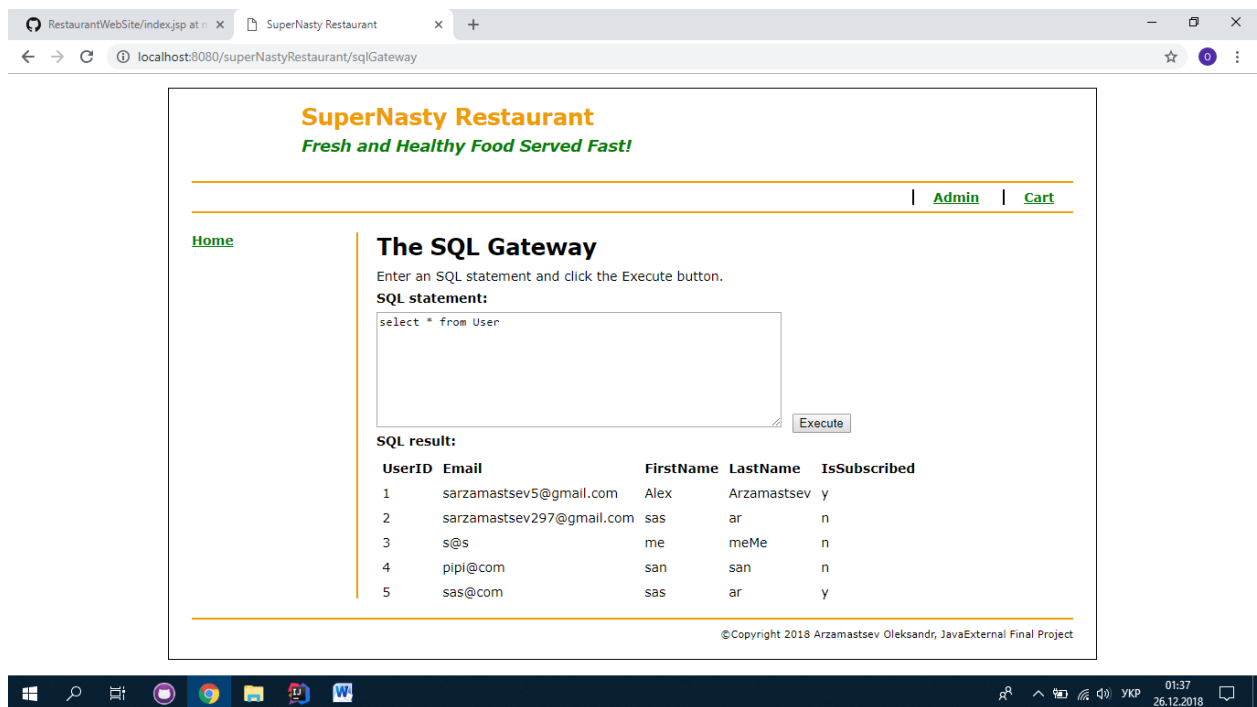


Рис. 16. SQLGateWay.

Адміністратор має змогу напряму працювати із базою даних через інструмент, що зображений на рисунку. В текстовому полі вводиться запит мовою SQL, натискаючи на кнопку Execute – отримуємо відповідний результат, що свідчить про успішність/неуспішність написаного запиту.

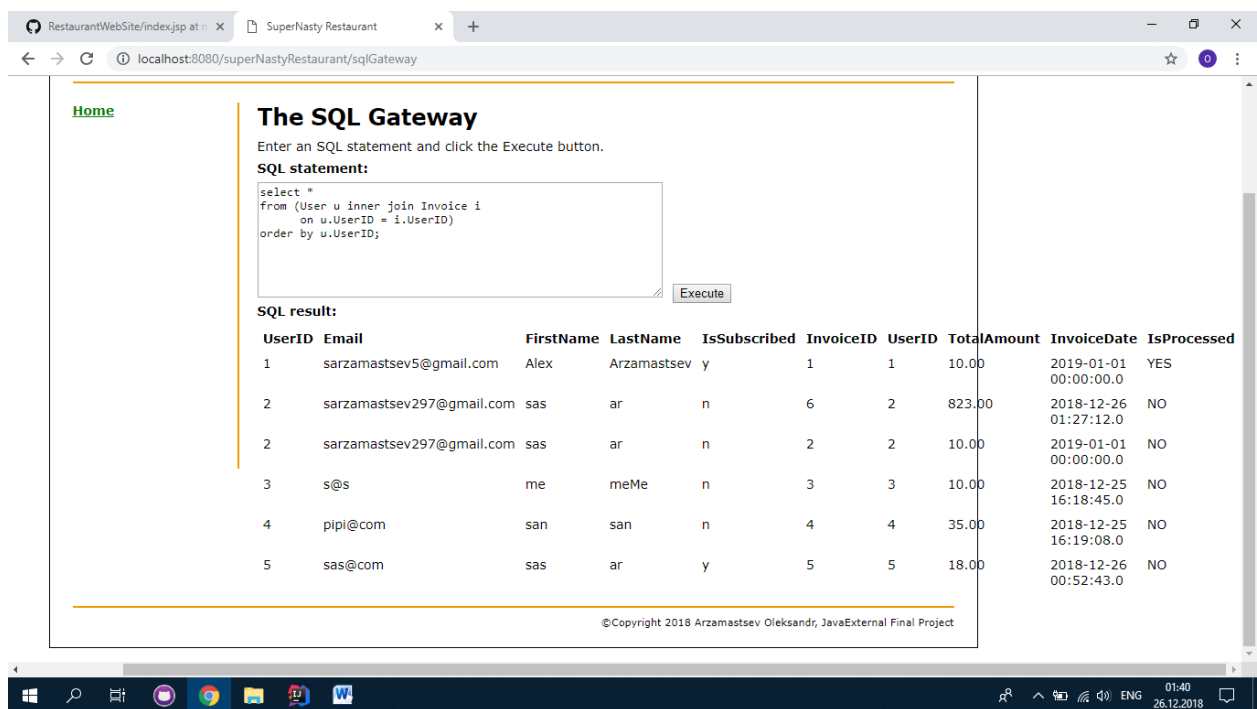


Рис. 17. Результат виконання SQL запиту.