



Co-funded by the  
Erasmus+ Programme  
of the European Union

**Establishing Modern Master-level Studies in Information Systems**  
**561592-EPP-1-2015-1- FR-EPPKA2-CBHE-JP**

# **MIS and Data Warehousing**

**Labs**

## Зміст

Лабораторна робота № 0. Введення в Oracle Apex. Схема авторизації.....	3
Лабораторна робота № 1. Створення проекту. Сторінка реєстрації та Home .....	9
Лабораторна робота № 2. Створення десктопного додатку .....	14
Лабораторна робота № 3. Основні оператори SQL. Створення бази даних, запити до даних, та форм для їх редагування .....	26
Лабораторна робота № 4. Основні оператори PL/SQL. Приклади реалізації бізнес-логіки .....	29
Лабораторна робота № 5. Основні оператори PL/SQL. Робота з SQL Developer. ....	33
Лабораторна робота № 6. Динамічні події в Apex. Структурний елемент “дерево” .....	49
Лабораторна робота № 7. Покращення користувацького інтерфейсу .....	54
Лабораторна робота № 10. Створення бота .....	75
Лабораторна робота № 11. Створення календаря та робота зі звітами .....	92

## Лабораторна робота № 0. Введення в Oracle Apex. Схема авторизації

Oracle Application Express (скорочено Oracle Apex, APEX) – вільне середовище швидкої розробки прикладного програмного забезпечення на основі СУБД Oracle Database, цілком реалізоване як веб-додаток (рис. 1).

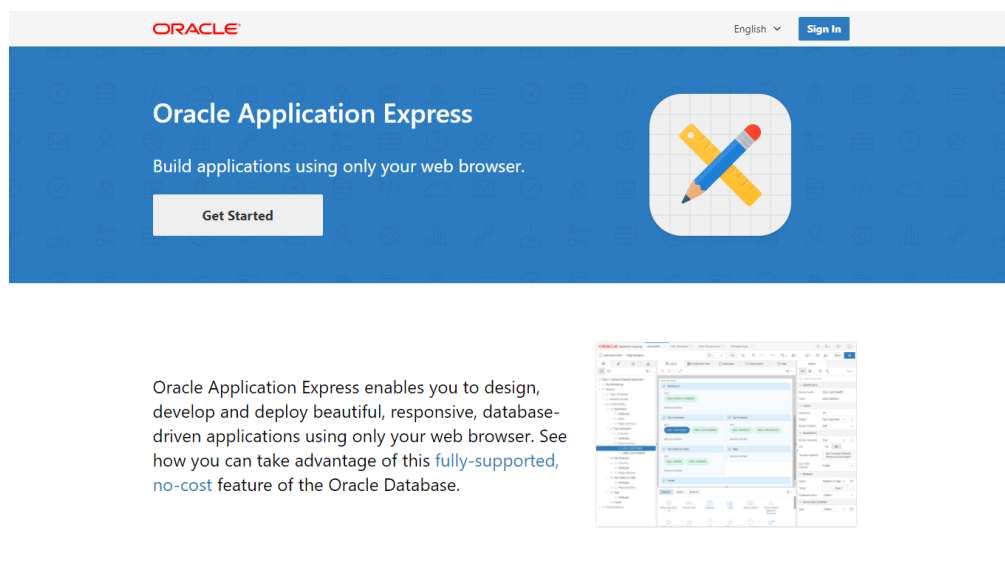


Рисунок 1 – Вигляд веб-додатку Oracle Apex

Для початку роботи з середовищем Oracle Apex необхідно перейти за посиланням [apex.oracle.com](https://apex.oracle.com) та пройти реєстрацію (рис. 2-3).

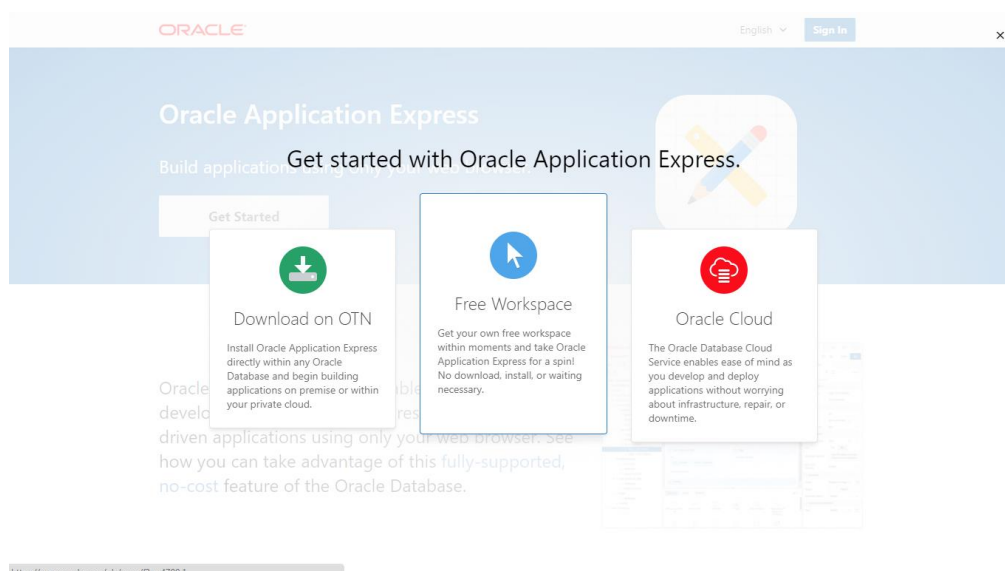


Рисунок 2 – Вибір безкоштовного робочого простору

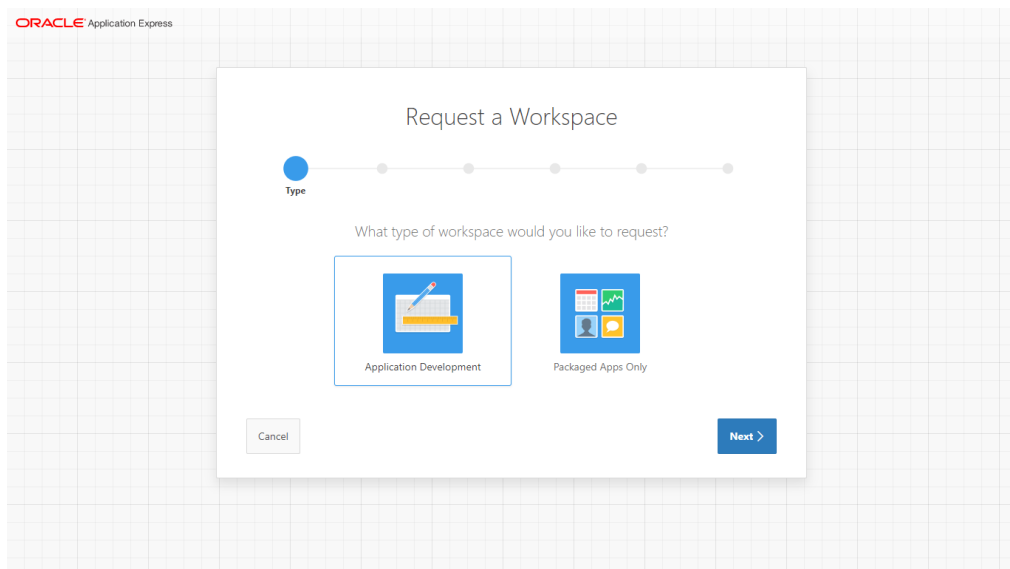


Рисунок 3 – Вибір типу робочого простору (розробка додатків)

Після завершення усіх кроків на пошту, вказану під час реєстрації, буде відправлено лист з посиланням на встановлення паролю до облікового запису.

Вигляд робочого простору показано на рисунку 4.

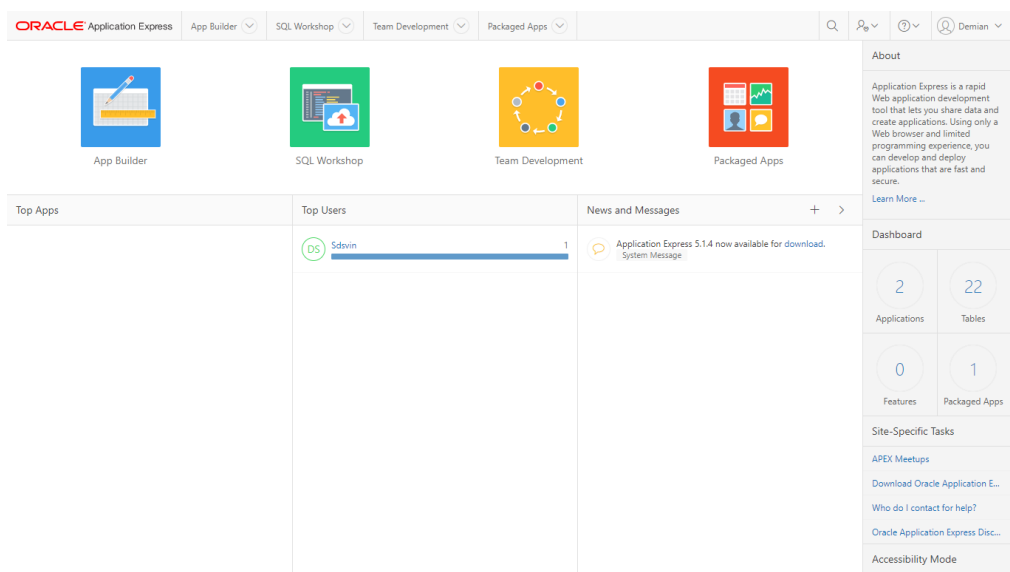


Рисунок 4 – Робочий простір Oracle Apex

На вкладці Packaged Apps (рис. 5) розміщені демонстраційні додатки для навчання, кожен з яких відображає певну область даного середовища (робота з графіками, формами і т. д.) для розробників.

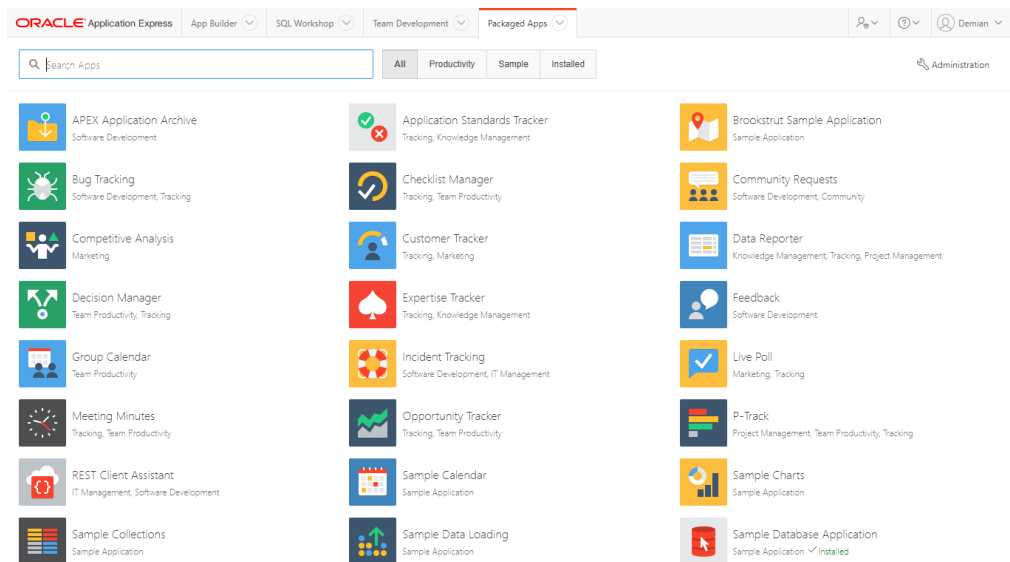


Рисунок 5 – Вкладка з демонстраційними додатками

Розглянемо на прикладі встановлення додатку Sample Database Application (рис. 6).

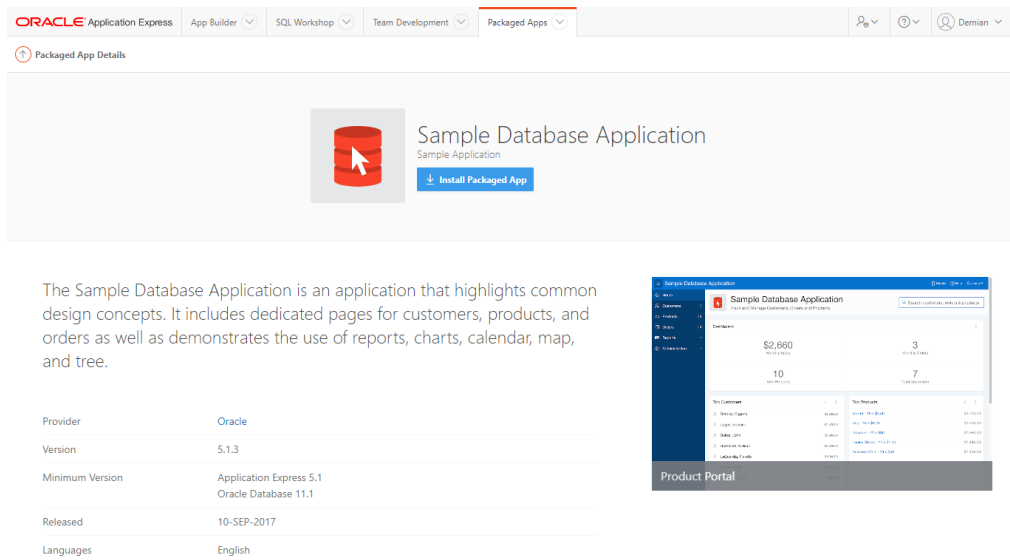




Рисунок 6 – Вікно встановлення додатку

Процес встановлення додатку дуже простий та займає мало часу. Після встановлення будь-якого додатку, він з'являється на вкладці App Builder (рис. 7), де його можна змінювати  чи запускати .

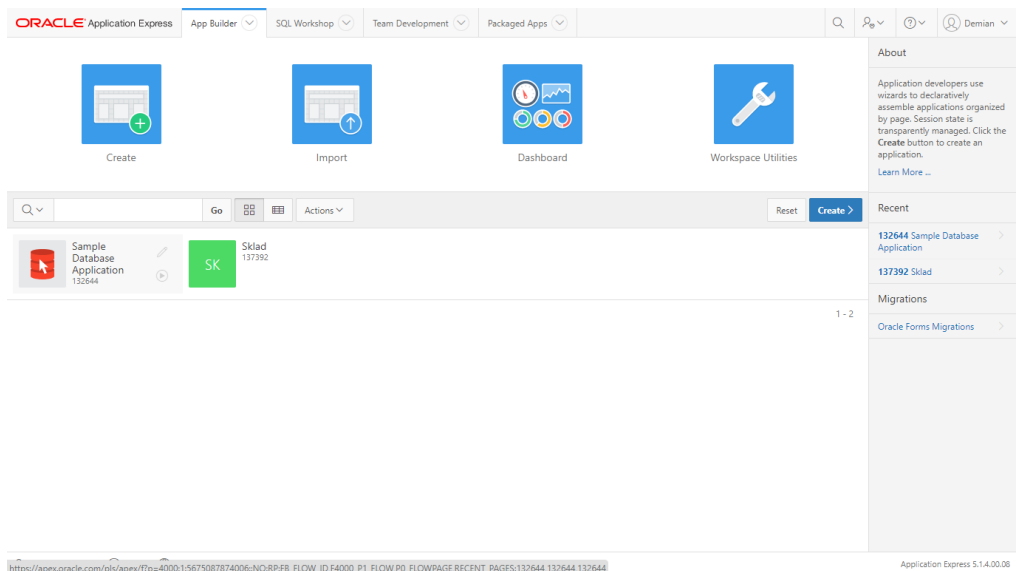


Рисунок 7 – Вкладка App Builder

Двічі натиснувши на додаток відбувається перехід на сторінку його редагування (рис. 8).

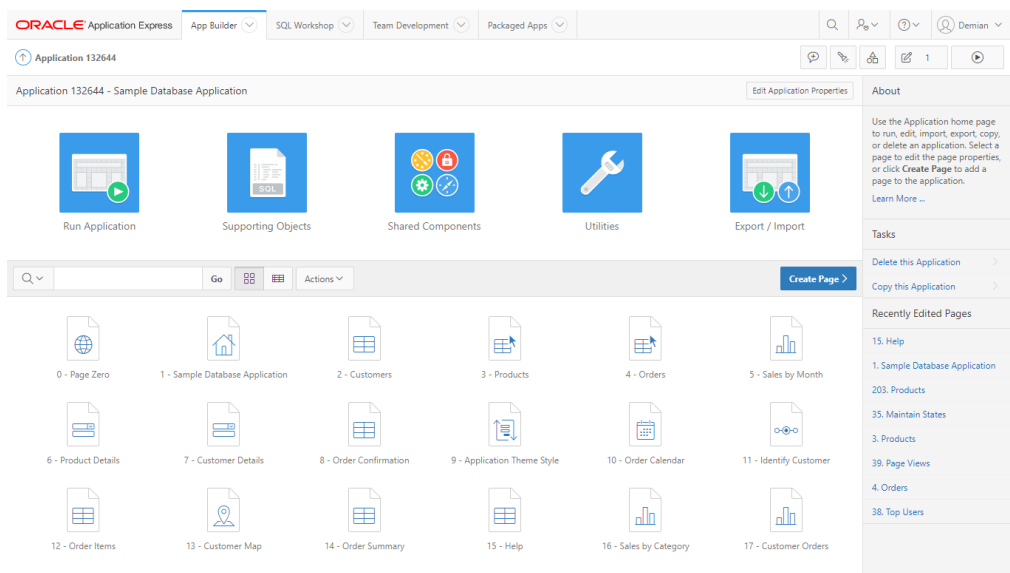


Рисунок 8 – Сторінка редагування додатку

Кожен додаток складається зі сторінок, які теж, відповідно, можна редагувати, двічі натиснувши на них. Відкривається панель управління для редагування сторінок (рис. 9), де повністю показана їх структура.

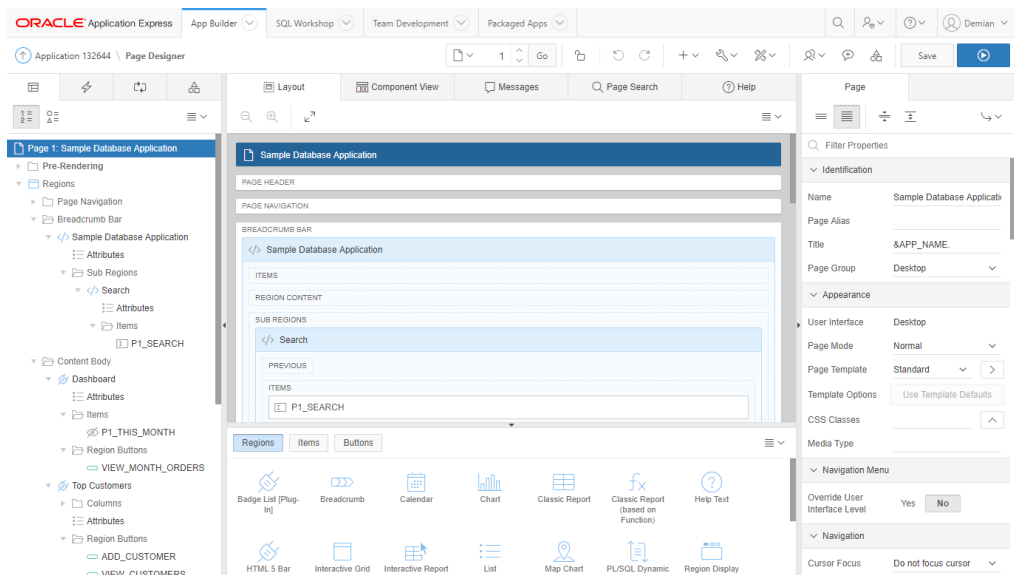


Рисунок 9 – Панель управління для редагування сторінок

Також, на вкладці App Builder є можливість створювати власні додатки (Create), імпортувати існуючі (Import), відкрити панель моніторингу додатків (Dashboard) та налаштовувати робочий простір (Workspace Utilities).

Вигляд запущеного додатку показано на рисунку 10.

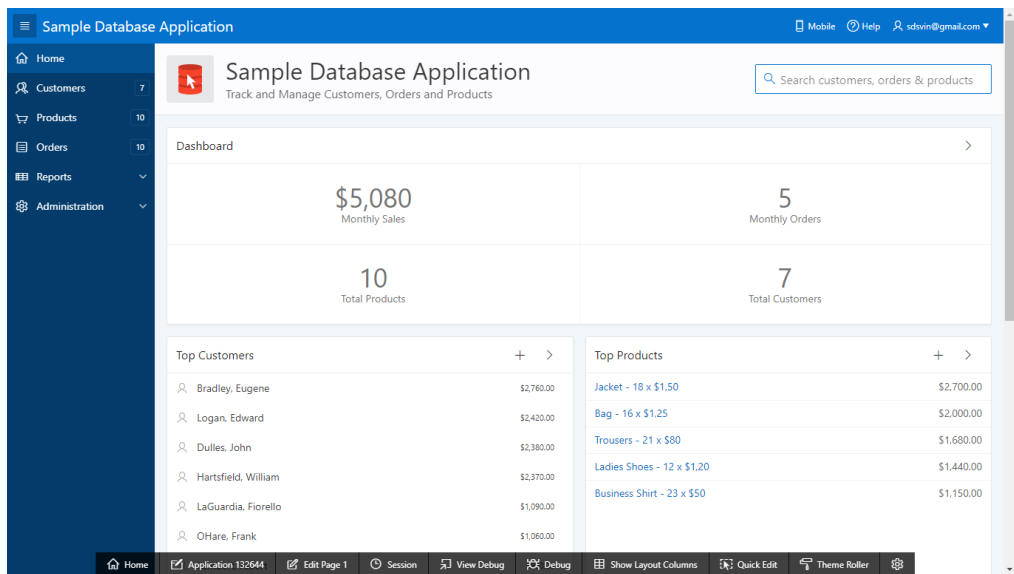


Рисунок 10 – Додаток Sample Database Application

На вкладці SQL Workshop (рис. 11) розміщені усі елементи управління для роботи з базами даних.

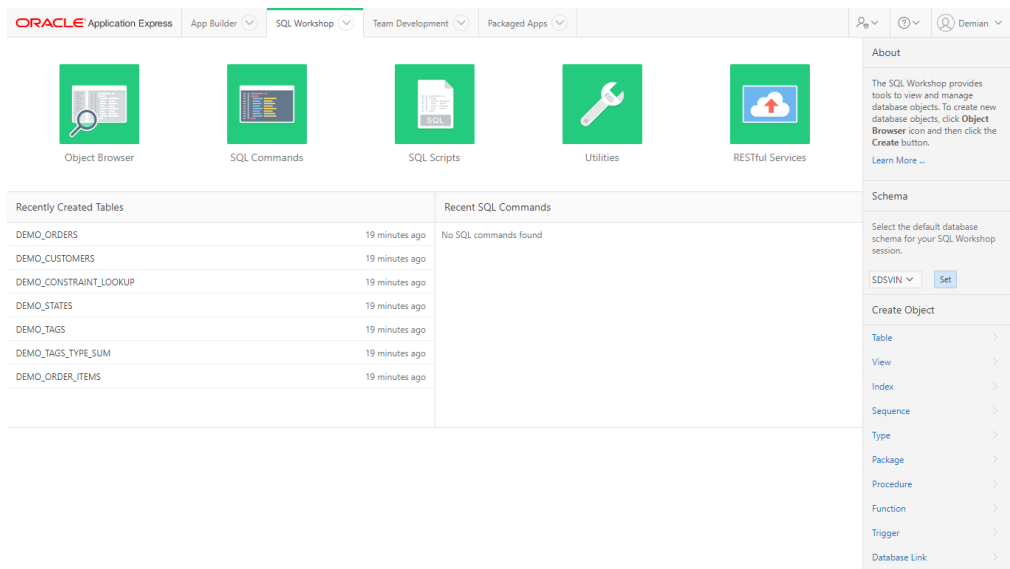


Рисунок 11 – Вкладка SQL Workshop для роботи з базами даних

Вкладка Team Development (рис. 12) призначена для командної розробки додатків.

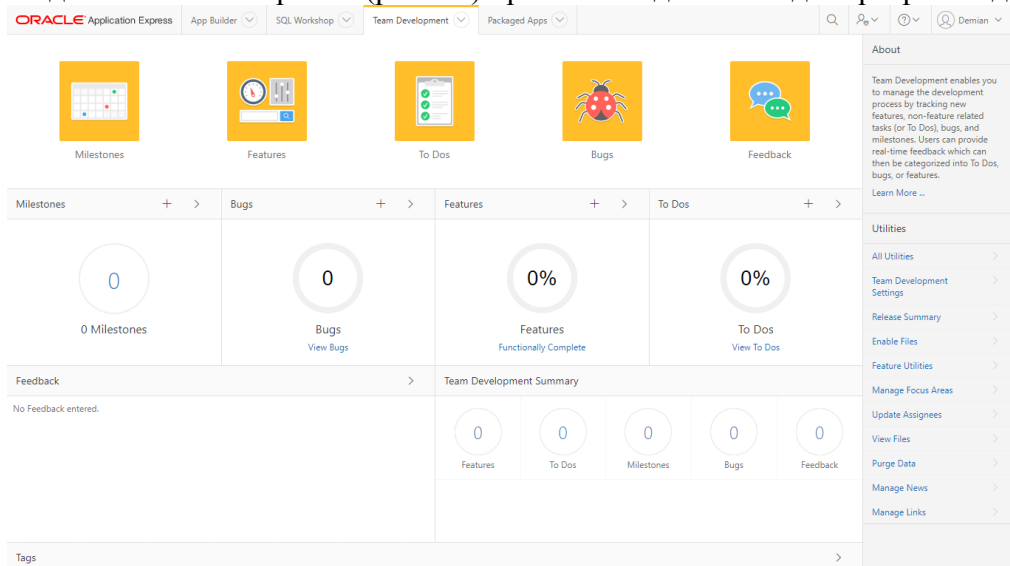


Рисунок 12 – Вкладка Team Development для командної розробки



## Лабораторна робота № 1. Створення проекту. Сторінка реєстрації та Home

Для роботи з користувачами необхідно створити таблицю. Для цього необхідно зайти на вкладку SQL Workshop. Щоб переглянути усі об'єкти, що містяться в базі даних, необхідно перейти на вкладку Object Browser (рис. 1).

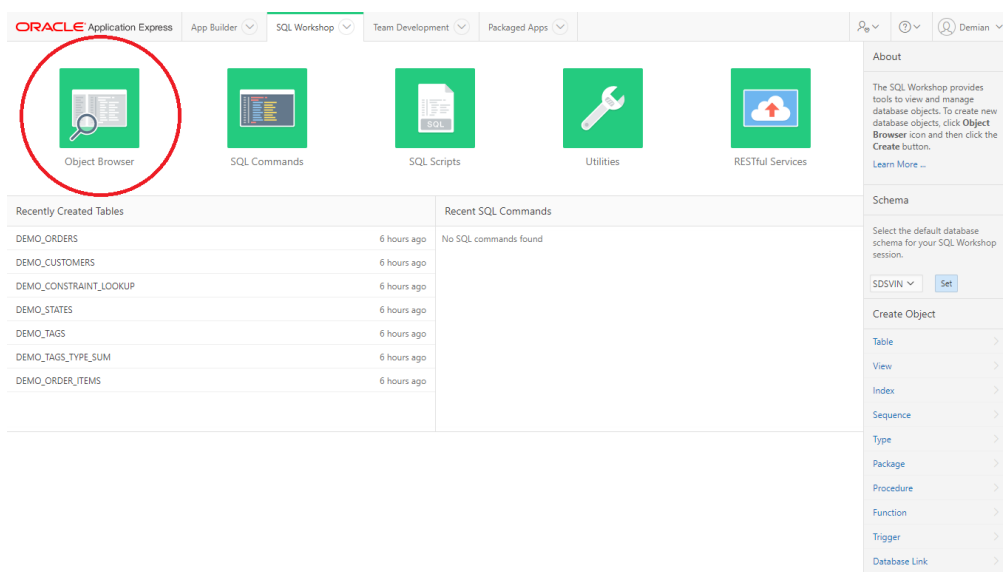



Рисунок 1 – Вкладка Object Browser

Для створення нової таблиці необхідно натиснути клавішу  та обрати Table (рис. 2).

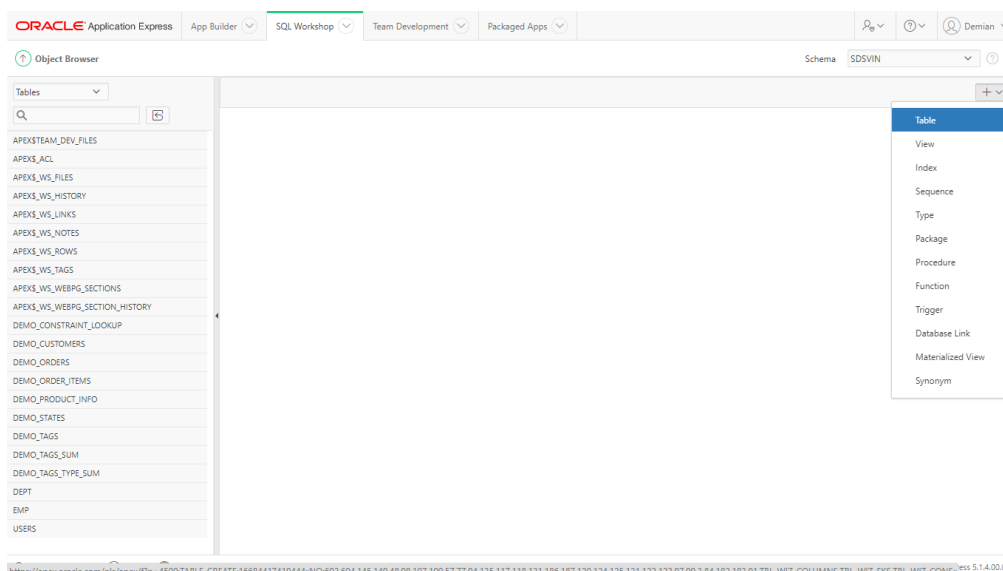


Рисунок 2 – Створення нової таблиці у базі даних

Для прикладу створюємо таблицю користувачів **USERS**, що буде містити наступні колонки:

- **id** – індекс користувача (тип даних: **NUMBER** – усі числа);
- **name** – ім'я користувача (тип даних: **VARCHAR2** – символи, розмір: 20).

В результаті структура майбутньої таблиці повинна мати вигляд як на рисунку 3.

Columns

\* Table Name:  ?

Preserve Case

Column Name	Type	Precision	Scale	Not Null	Identity	Move
<input type="text" value="id"/>	NUMBER	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	- None -	^ v
<input type="text" value="name"/>	VARCHAR2		<input type="text" value="20"/>	<input type="checkbox"/>		^ v
<input type="text"/>	- Select Datatype -					^ v
<input type="text"/>	- Select Datatype -					^ v
<input type="text"/>	- Select Datatype -					^ v
<input type="text"/>	- Select Datatype -					^ v
<input type="text"/>	- Select Datatype -					^ v
<input type="text"/>	- Select Datatype -					^ v

Рисунок 3 – Структура таблиці користувачів

На наступному кроці (рис. 4) необхідно вибрати ключове поле – індекс. Для цього необхідно значення Primary Key встановити на Populated from a new sequence і обрати необхідну колонку ID(NUMBER).

Primary Key

Table name:  ?

Primary Key:  No Primary Key ?  
 **Populated from a new sequence**  
 Populated from an existing sequence  
 Not populated  
 Populated by Identity column

\* Primary Key Constraint Name:  ?

\* Primary Key:  ?

\* Sequence Name:  ?

Рисунок 4 – Встановлення ключового поля

На наступних кроках створення таблиці можна налаштовувати в ній зв'язки (рис. 5) та унікальність рядків. Для того, щоб серед користувачів не повторювалися імена, потрібно у полі Constraint Type обрати Unique, а у полі Key Column(s) перемістити NAME у праву колонку та натиснути клавішу Add. В результаті повинно вийти як на рисунку 6.

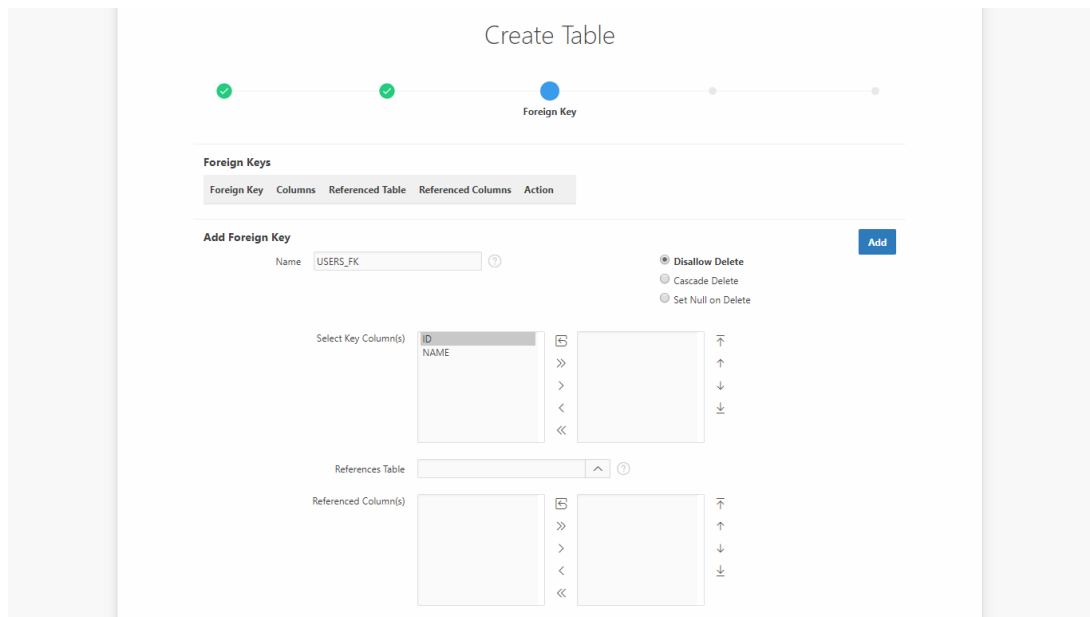


Рисунок 5 – Налаштування зв'язків у таблиці

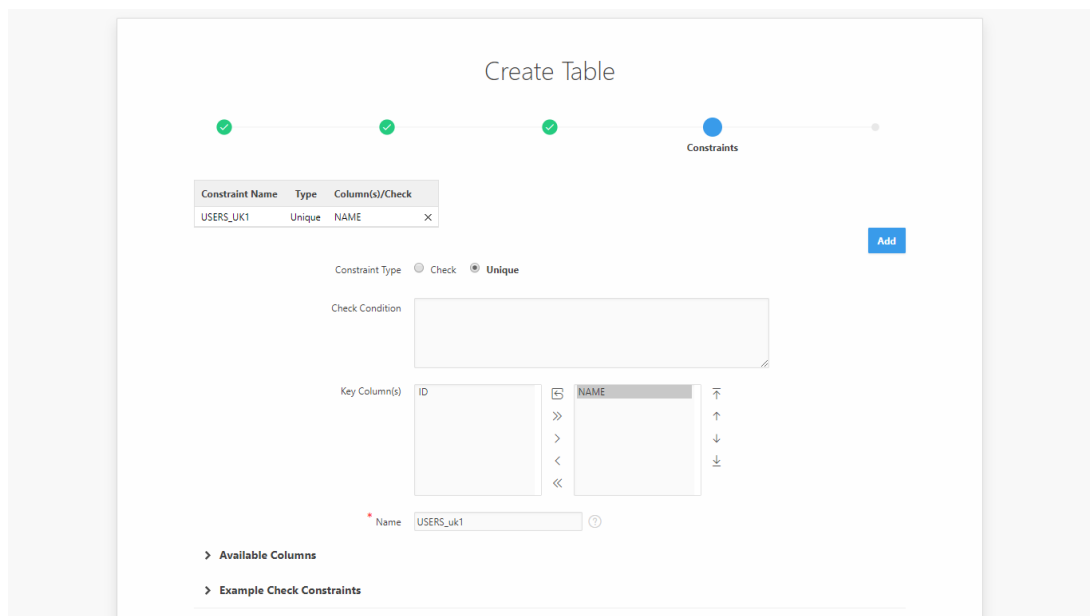


Рисунок 6 – Встановлення унікальності для колонки NAME

На останньому кроці можна побачити як створена таблиця виглядає на мові SQL (рис. 7).

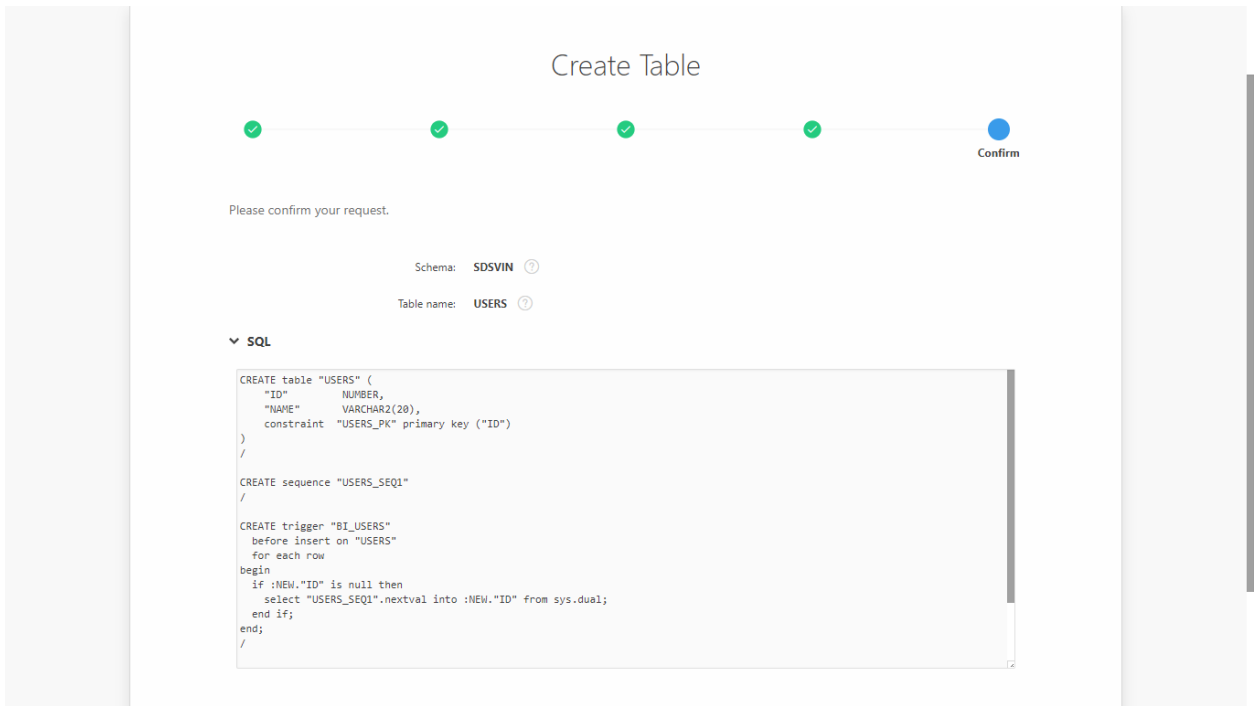


Рисунок 7 – Вигляд таблиці USERS на мові SQL

Сторінка додатку в Oracle APEX поділяється на регіони (рис. 8).

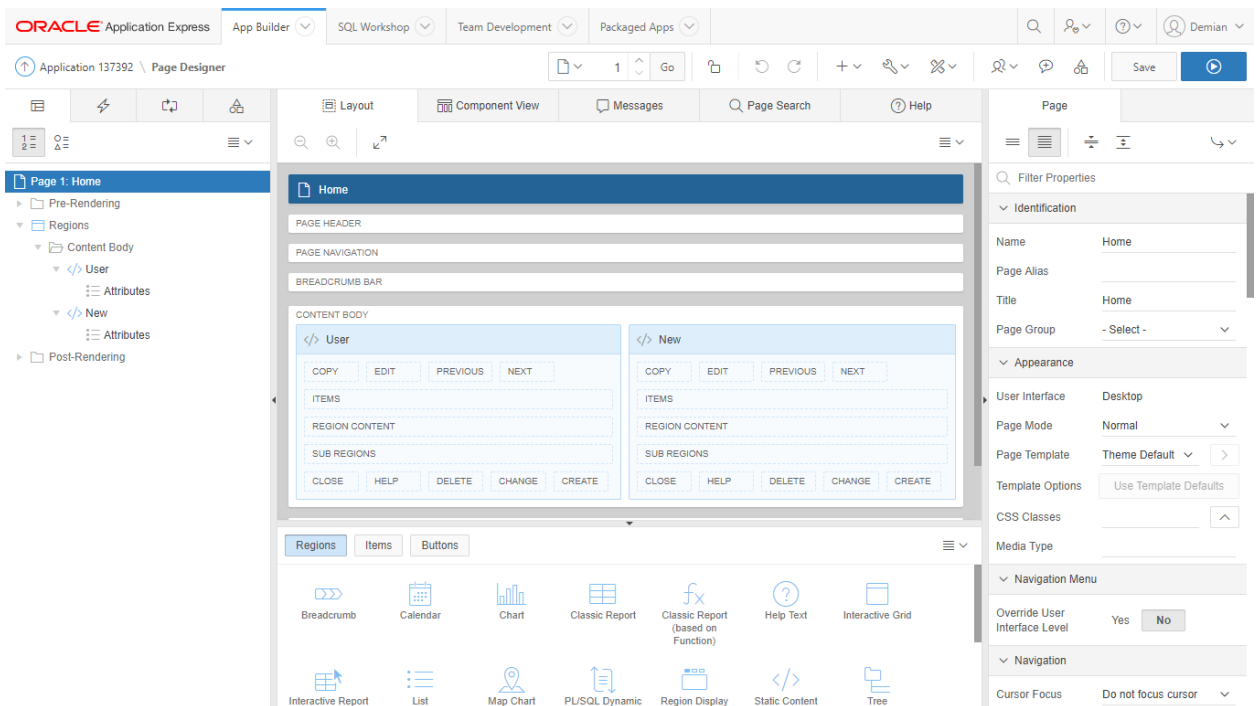


Рисунок 8 – Поділ сторінки додатку на регіони

Для додання авторизації необхідно налаштувати схему автентифікації. Необхідно обрати додаток, перейти у розділ Shared Components та обрати пункт Authentication Schema (рис. 9).

ORACLE Application Express App Builder SQL Workshop Team Development Packaged Apps

Application 137392 \ Shared Components \ Authentication Schemes

Authentication Schemes Subscription History

Q Go Actions Reset Create

Name	Scheme Type	Subscribed From	Subscribers
APEX	Application Express Accounts		
Custom User - Current	Custom		

1 - 2

An Authentication Scheme is saved configuration you can apply to your application. To turn off authentication you must create and make current a **No Authentication** scheme.

sdsvin@gmail.com sdsvin en Copyright © 1999, 2017, Oracle. All rights reserved. Application Express 5.1.4.00.08

Рисунок 9 – Налаштування схеми автентифікації

Для додання користувацької схеми автентифікації необхідно використати функцію (рис. 10).

Source

PL/SQL Code ?

```

1 function ischecked(
2     p_username IN VARCHAR2,
3     p_password IN VARCHAR2)
4 RETURN BOOLEAN IS
5 BEGIN
6
7     RETURN true;
8 END;
```

Рисунок 10 – Функція для користувацької схеми автентифікації

## Лабораторна робота № 2. Створення десктопного додатку

Для того щоб почати створювати десктопний додаток потрібно перейти у «App Builder» та обрати пункт меню «Create» або натиснувши кнопку рис. 1.

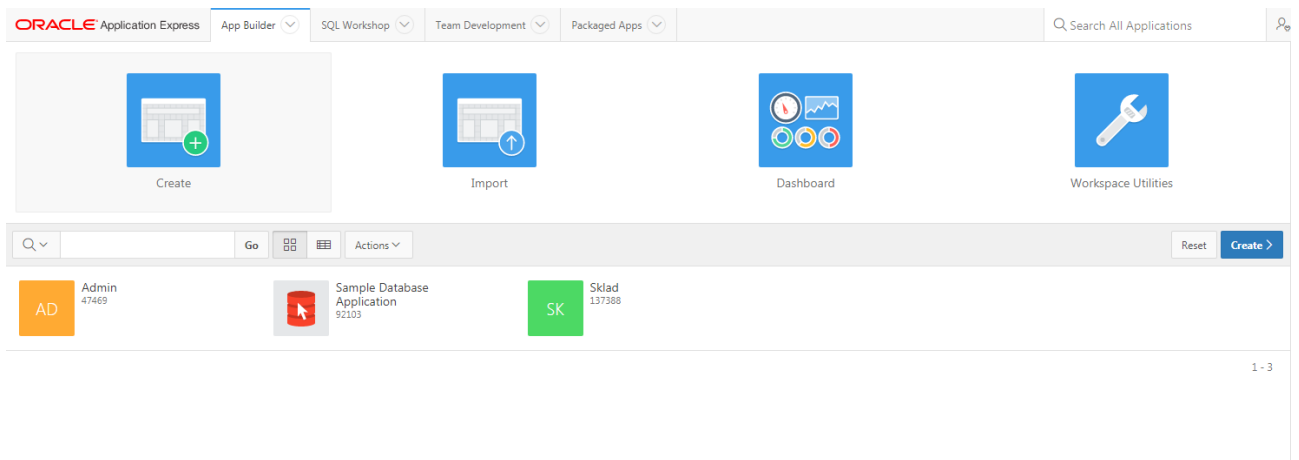


Рисунок 1 – Вигляд розділу «App Builder»

Після натискання кнопки «Create», Oracle запропонує створити окрім потрібного нам десктопного додатку ще мобільний додаток та веб-додаток, та пустий проект «Blueprint» рис. 2.

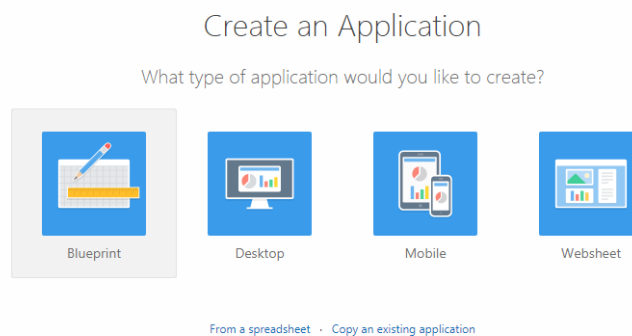


Рисунок 2 – Меню для створення додатку

### Десктопні та мобільні додатки для баз даних

Засоби бази даних дозволяють розробникам повністю контролювати всі аспекти процесу розробки та остаточної функціональності додатків. За допомогою додатків баз даних розробники можуть безпосередньо використовувати свої навички програмування SQL та PL / SQL. Програми баз даних використовують декларативні елементи керування для розробки, покращення та підтримки додатків. Вони також підтримують повне налаштування користувальницького інтерфейсу за допомогою шаблонів і тем.

При створенні додатка для комп'ютерів, інтерфейс користувача оптимізований для більшої кількості дисплеїв, таких як настільні, ноутбуків та планшети. Універсальна тема надає функції, які дозволяють масштабувати вміст сторінки різним розмірам екрана та ефективно використовувати доступний розмір екрана.

Під час створення додатків для мобільних баз, користувальницький інтерфейс, насамперед, орієнтований на менші екрани, як на смартфонах. Використовуючи функції, такі як таблиця reflow, перемикач стовпців та панелі, вміст сторінки збільшується і для використання на більших пристроях. Використовуючи систему jQuery Mobile, додатки для мобільних телефонів реагують на мобільні конкретні події, такі як зміна дотику та орієнтації.

## Веб-додатки

Веб-додатки дозволяють користувачам створювати програми, орієнтовані на дані, без будь-яких знань про програмування SQL. Веб-додатки - це спрощені програми, які підтримують сторінки, сітки даних і звіти. Ці додатки дуже прості у складанні та призначені для підтримки участі громади.

Обираємо блок «desktop». Створення додатку складається з наступних 5 кроків:

- 1) Введення початкових параметрів додатку. Схема, з якою він буде працювати, назва, номер, тему та її стиль. Номер додатку краще не змінювати, та як він повинен бути унікальним. Тема «Universal Theme» є адаптивною і добре підходить для різних екранів. Всі інші створені «табличній» верстці рис 3.

Create an Application

Name

User Interface Desktop

Schema BRUCESHEMA

Name Insanefury31 1

Application 58457

Theme Universal Theme (42)

Theme Style Vita (Accessibility Tested)

< Cancel Create Application Next >

Рисунок 3 – Задання початкових параметрів

- 2) Додавання початкових сторінок рис 4.

Create an Application

Pages

Page	Name	Type	Page Mode	Source Type	Source	Parent Page	
	1	Home	Blank	Normal	-	-	- x

Add Page

< Cancel Create Application Next >

Рисунок 4 – Додавання сторінок

- 3) Додавання спільних компонентів. Спільні компоненти - це звичайні елементи програми, які можуть відображатися або застосовуватися на кількох сторінках у програмі. Щоб заощадити час або підтримувати узгодженість між додатками рис. 5.

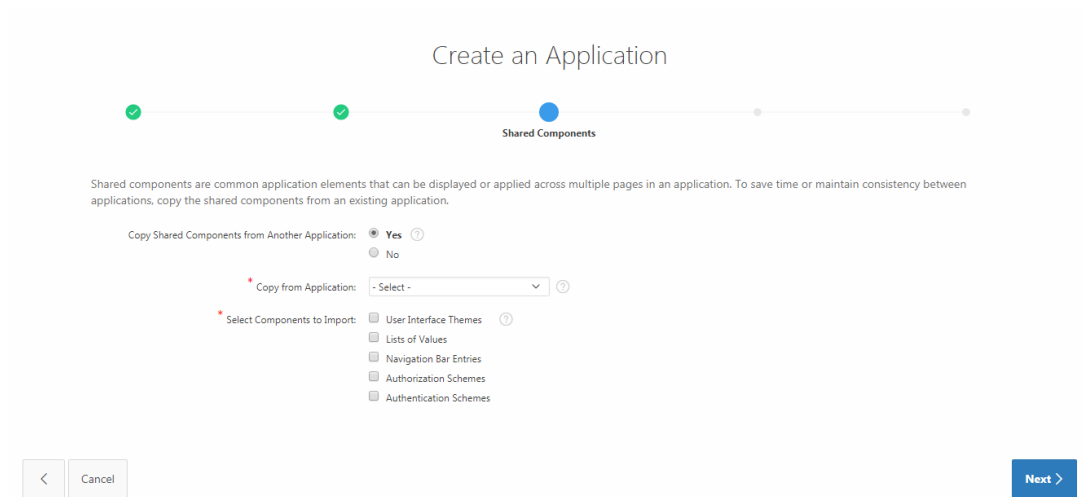


Рисунок 5 – Додавання спільних компонентів

- 4) Додавання атрибутів додатку. Полягає у додаванні мови додатку та визначення у якому вигляді буде відображатись дата рис. 6.

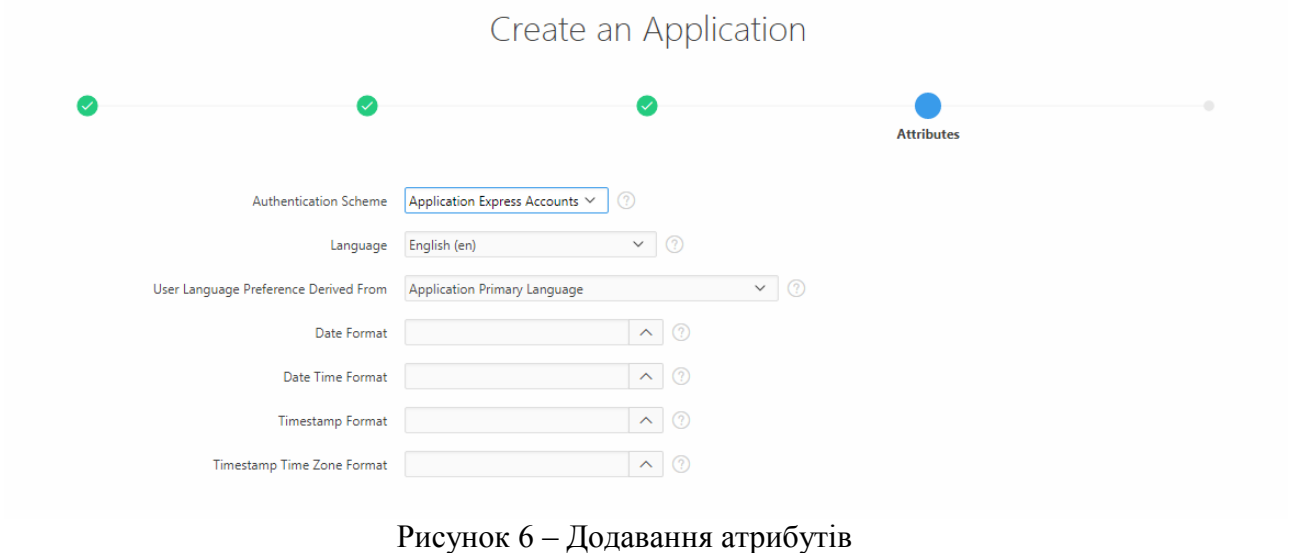


Рисунок 6 – Додавання атрибутів

- 5) Підтвердження попередньо введених параметрів та атрибутів рис. 7.



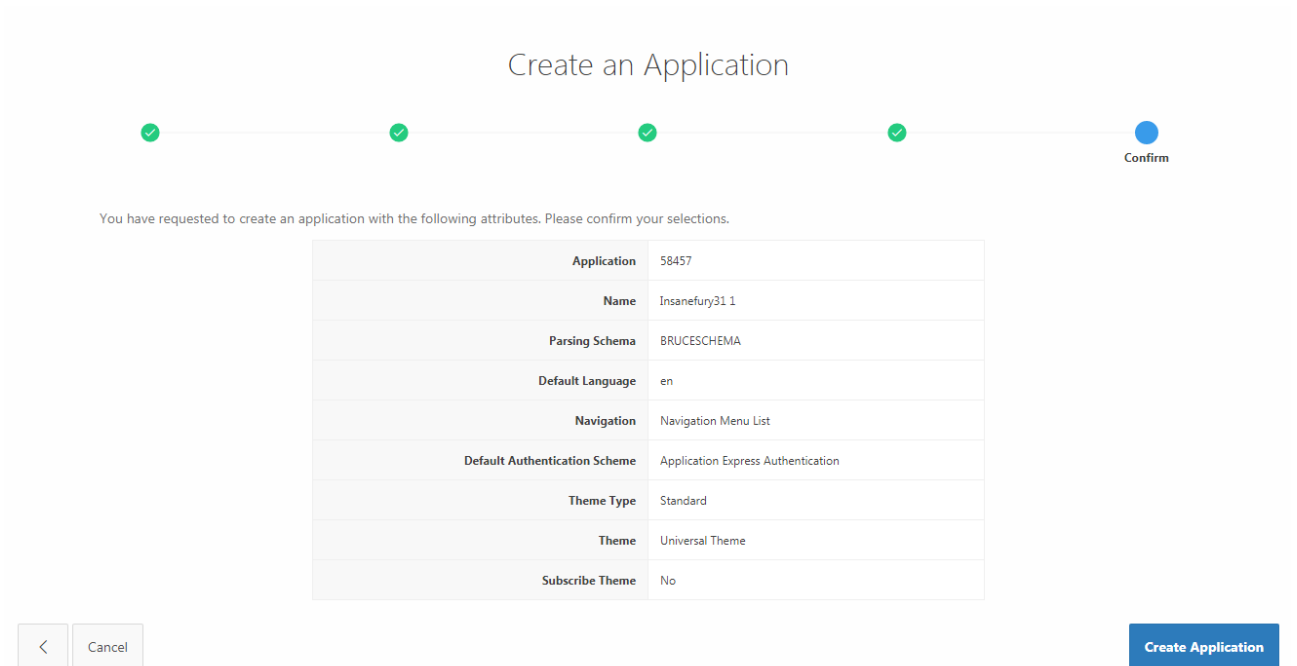


Рисунок 7 – Підтвердження створення додатку

Після створення додатку, його можна буде побачити у «App Builder» рис. 8.

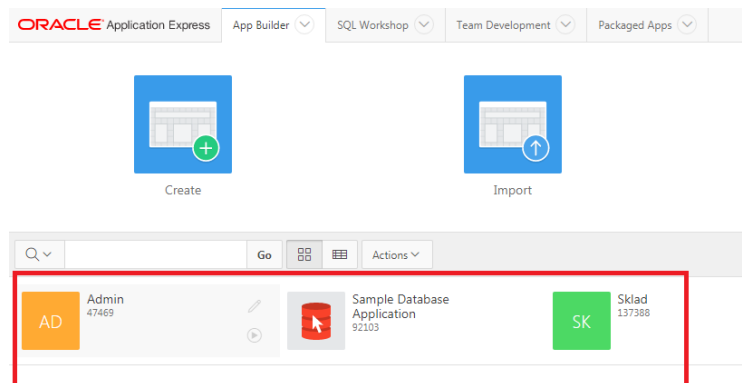
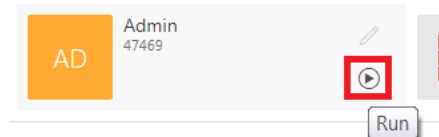
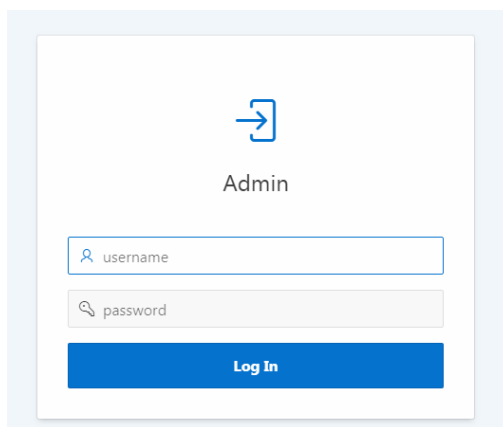


Рисунок 8 – Список створених додатків

Для того щоб зайти у наш створений додаток, потрібно натиснути «Run» на його блоці,



та ввести наш username та password при реєстрації на сайті apex.oracle.com.



Після вдалої авторизації в додатку, повинно бути наступна сторінка рис. 9.

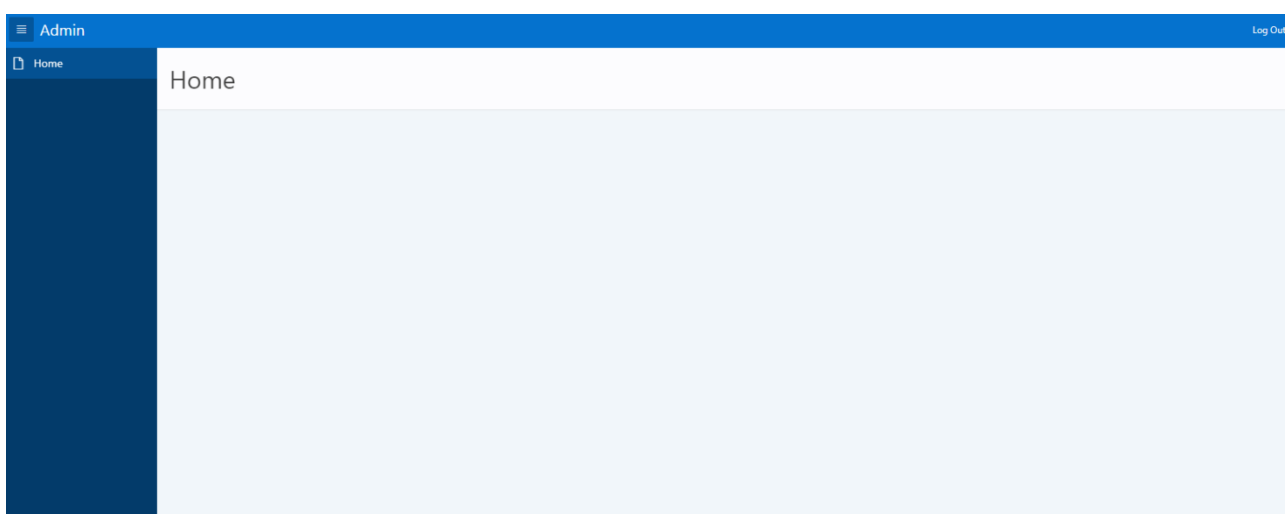


Рисунок 9 – Вигляд створеного додатку

### Додавання сторінки для редагування користувачів

Для того щоб додати сторінку, потрібно повернутись у «App Builder», натиснути на іконку нашого створеного додатку, та натиснути на кнопку «Create Page» рис. 10.

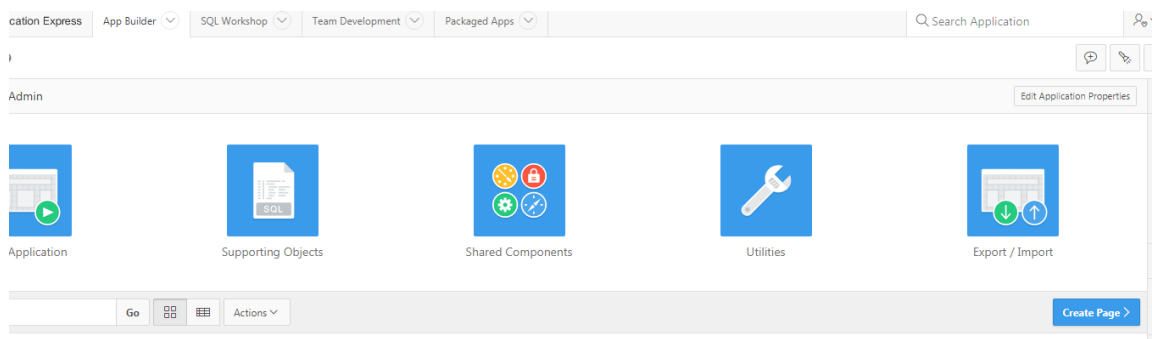


Рисунок 10 – Меню редактора додатку

У редакторі створенні сторінки, буде вибір з декількох стандартних майстрів сторінок, але нам потрібний тип «Form» рис. 11. Детального опису кожного з майстрів можна знайти натиснувши на кнопку «Help».

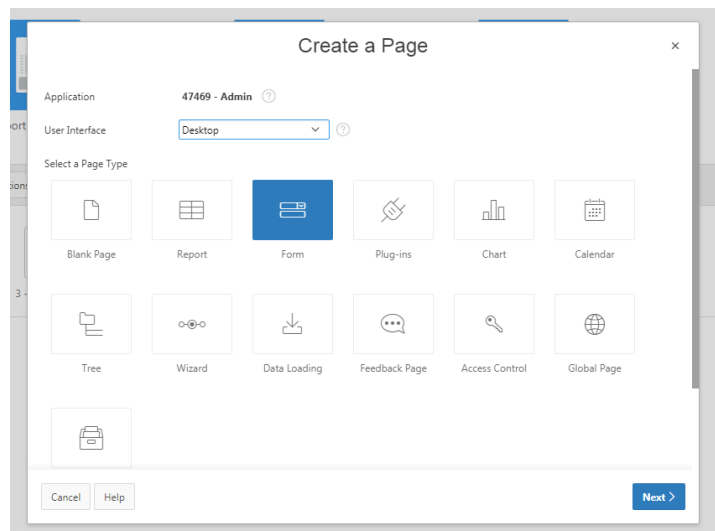


Рисунок 11- Набір майстрів сторінок

Далі будуть більш спеціалізовані майстри для типу «Form» рис 12. Обираємо «Report with Form on Table».

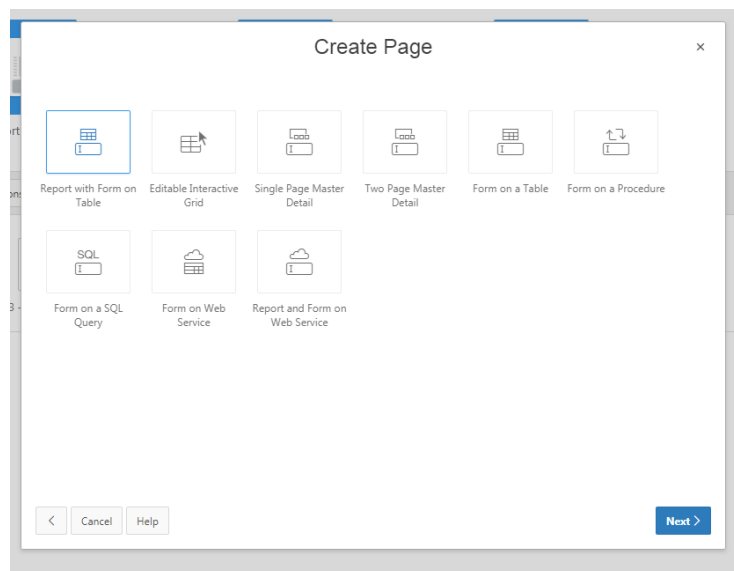


Рисунок 12- Набір майстрів для типу «Form»

Далі відкриється вікно для ведення початкових параметрів сторінки. Воно буде складатись з 4 кроків:

1) Report Type(вид списку):

- Interactive Grid (створює інтерактивну сітку для читання);
- Interactive Report (створює інтерактивний звіт);
- Classic Report (створює класичний звіт на основі шаблону).

Номера назви сторінок для виведення списку та редагування запису.

Модель сторінки визнає режим сторінки на основі наступних параметрів:

- Normal: сторінка представлена як звичайна сторінка додатків Application Express.
- Modal Dialog: сторінка представлена як модальне діалогове вікно. Модальне діалогове вікно - це оверлеяне вікно, розташоване у вікні перегляду, який залишається активним і сфокусованим, доки кінцевий користувач не відхилить (закриє) його. Нижня сторінка неактивна, і кінцевий користувач не може взаємодіяти з іншою частиною сторінки, доки діалогове вікно не буде закрито.

Зверніть увагу, що опція «Modal Dialog» доступна лише для вибору, де поточна тема програми містить принаймні один шаблон сторінки діалогового вікна типу шаблон.

Breadcrumb: Визначає чи потрібно використовувати навігаційну панель на панелі навігації на вашій сторінці та яку панель керування навігаційною клавішею ви хочете використовувати.

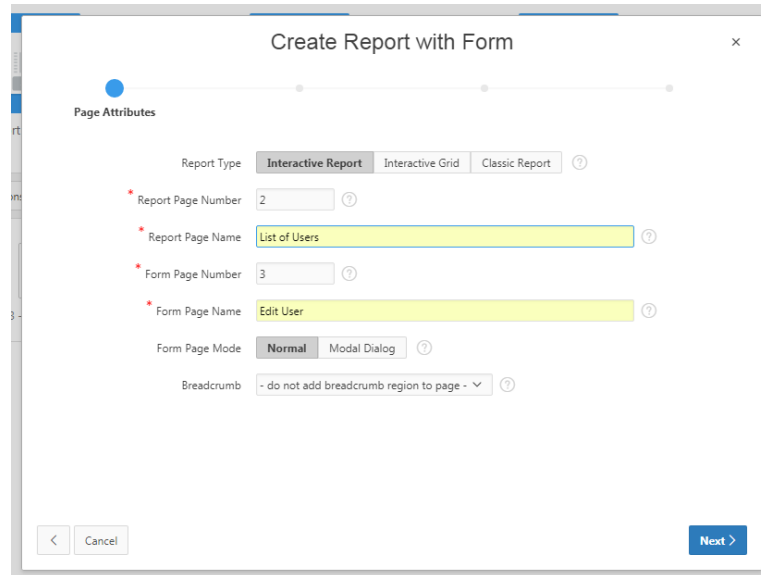


Рисунок 13 – Введення початкових параметрів сторінок

- 2) Додавання сторінки у навігаційне меню.
- 3) Задаємо, які саме поля можна буде переглядати у списку рис. 14.

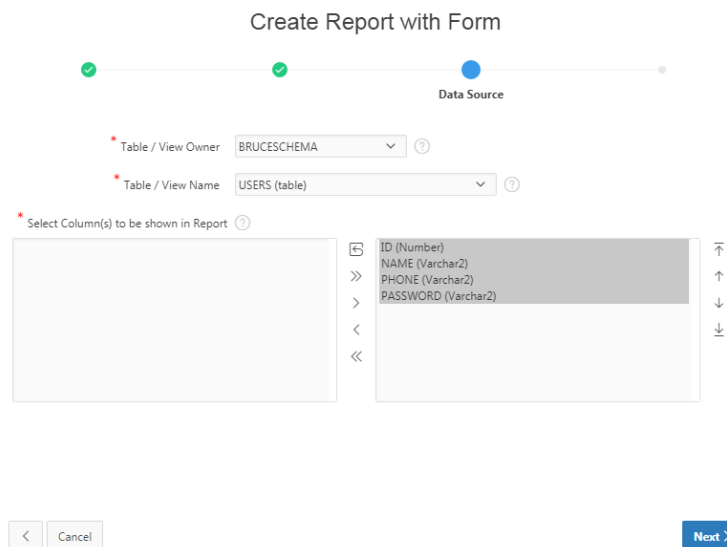


Рисунок 14 – Вибір схеми, таблиці та параметрів, які будуть зображені у списку

- 4) Задаємо, які саме поля можна буде редагувати рис. 15.  
Виберіть «Managed by Database» (ROWID), щоб мати форму, використовуючи псевдосполучення ROWID, щоб визначити рядки для оновлення та видалення. Виберіть «Select Primary Key Column(s)» для використання стовпців основних ключових слів джерельної таблиці. Обираємо 2 варіант. Задаємо ключ по якому буде виконуватись редагування. Задаємо джерело нашого ключа:

- Existing Trigger(існуючий тригер), якщо існуючий тригер буде використано для заповнення первинного ключа або якщо користувачеві буде потрібно ввести значення основної клавіші вручну.
- Existing Sequence(існуючу послідовність), якщо існуючу послідовність буде використано для створення первинного ключа.

Обираємо перший варіант.

Рисунок 15 – Обрання колон для редагування у формі

Після створення сторінки та відкривши її натиснувши на її іконку, запускаємо кнопкою «Run» рис. 16.

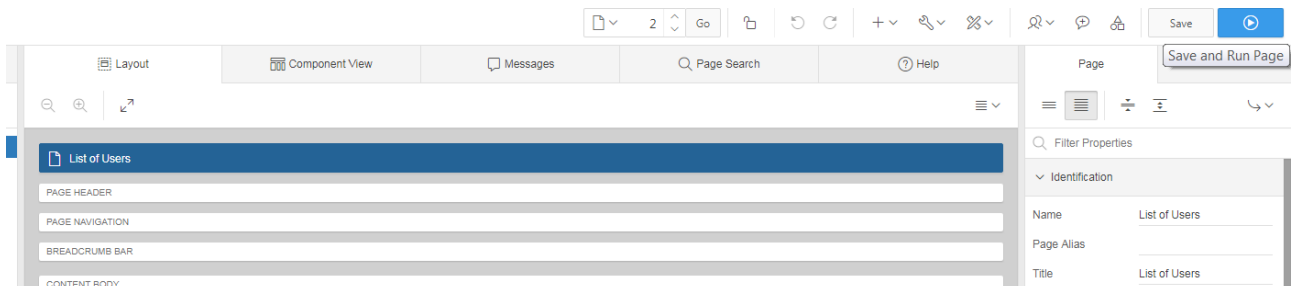


Рисунок 16 – Кнопка запуску сторінки.

Після запуску сторінки, має відкритись сторінка з наступним вмістом рис 17.

Id	Name	Phone	Password
1	test	-	123
21	test2	-	123

Рисунок 17 – Вигляд таблиці

Натиснувши на олівець, можна редагувати відповідне поле у таблиці рис 18.

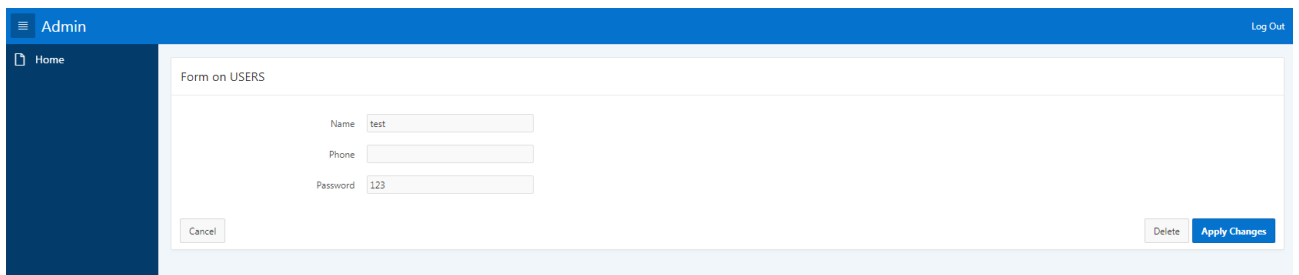


Рисунок 18 – Вигляд сторінки для редагування

Для того щоб зробити редагування більш інтерактивним, зробимо нашу сторінку редагування «Модальною». Для цього перейдемо до редагування сторінки. В пункті «Appearance» у атрибуті «Page Mode» оберемо «Modal Dialog» рис. 19.

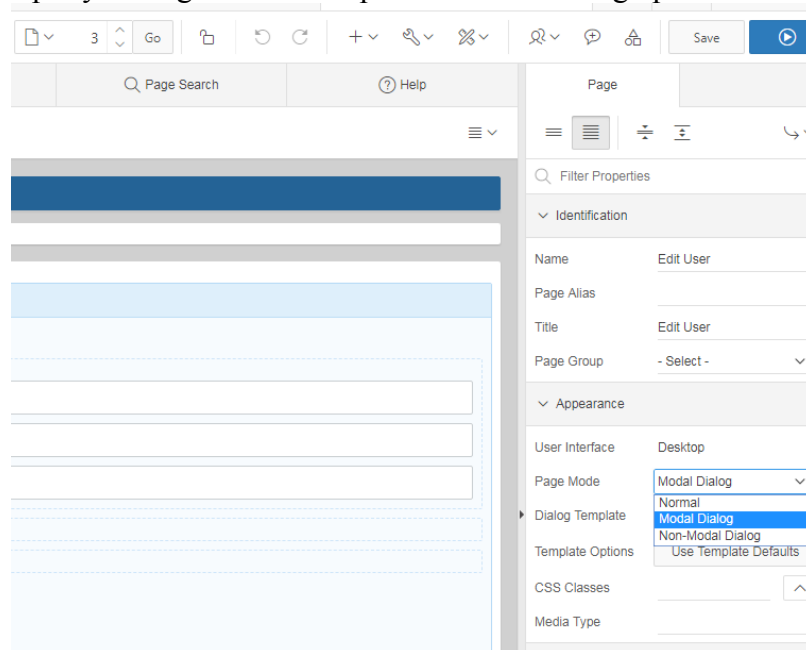


Рисунок 19 – Налаштування режиму сторінки

Після налаштування, наша сторінка буде відкриватись у «Модальному режимі» рис.20.

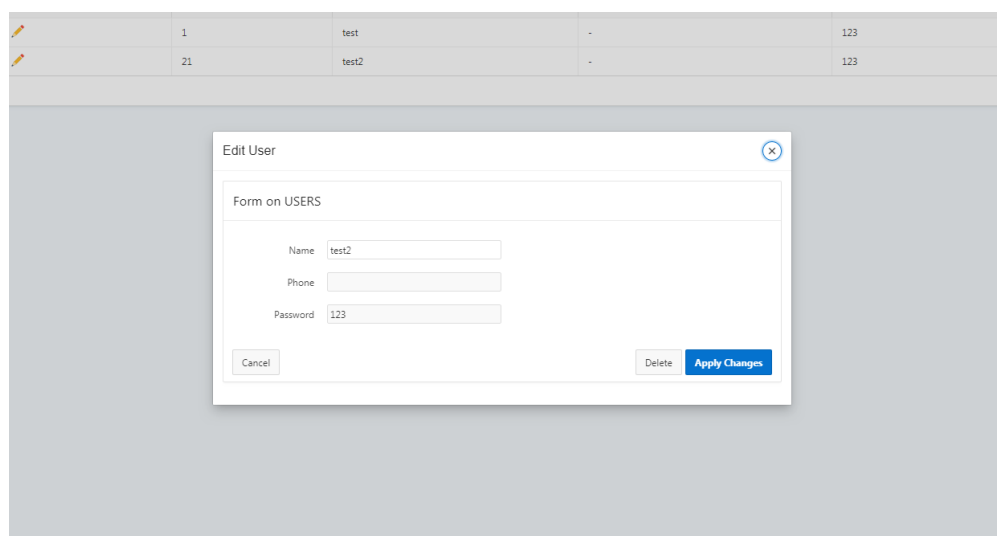


Рисунок 20 – Вигляд сторінки редагування у «Модальному режимі»

Додаємо довідку до елементів сторінки.

Для того щоб додати довідку то елемента, виберіть цей елемент у редакторі сторінки з її структури рис. 21.

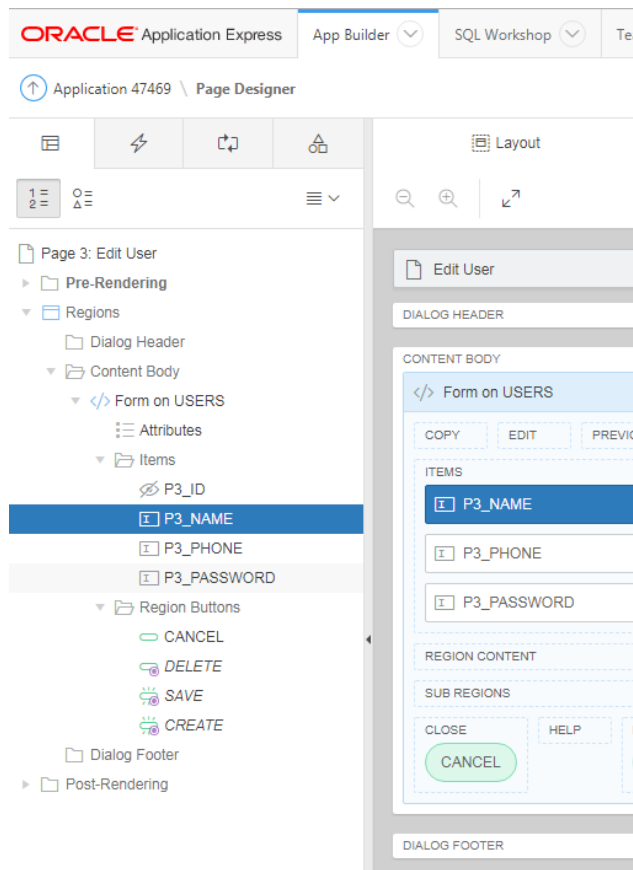


Рисунок 21 – Структура сторінки

У кожного елемента є параметри. Відкривши всі параметри рис. 22

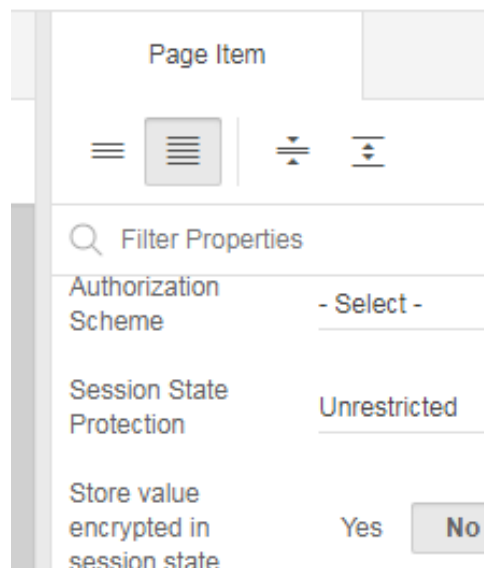


Рисунок 22 – Параметри сторінки

Переходимо до пункту «Help», де є два поля. Перше поле призначене для короткого опису під елементом, а другий – для більш глибокого пояснення рис. 23.

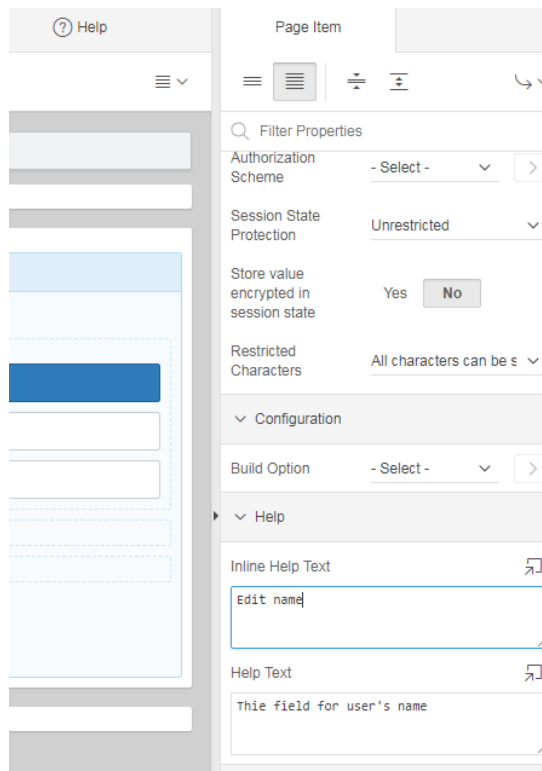


Рисунок 22 – Поля для введення довідки

Після збереження усіх змінених параметрів, на нашій сторінці редагування буде видно на рис 23. та рис 24.

The image shows a modal window titled 'Edit User' with a close button in the top right. Inside the modal, there is a section titled 'Form on USERS'. It contains three input fields: 'Name' with the value 'test', 'Phone' (empty), and 'Password' with the value '123'. Each input field has a small circular help icon to its right. Below the input fields, there are three buttons: 'Cancel', 'Delete', and 'Apply Changes'.

Рисунок 23 – Результат додавання довідки



Form on USERS

Name

Phone

Password  ?

Рисунок 24 – Вигляд додаткової довідки

## Лабораторна робота № 3. Основні оператори SQL. Створення бази даних, запити до даних, та форм для їх редагування

Для роботи з БД Oracle доцільно використовувати SQLDeveloper - інтегроване середовище розробки на мовах SQL і PL / SQL, з можливістю адміністрування баз даних, що орієнтована на застосування в середовищі Oracle Database.

Завантажити SQLDeveloper можна за посиланням:

<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

Так як для підключення до БД Oracle за допомогою SQLDeveloper необхідно купувати місце на платному сервері, розглянемо основні моменти стосовно мови PL/SQL.

### 1. Основи синтаксису

Основний програмно-лексичної одиницею мови є логічний блок, який може являти собою процедуру, функцію або анонімний блок. Кожен блок PL / SQL може включати до трьох секцій: описову або декларативну (ця секція необов'язкова), секцію виконуваного коду (Обов'язкова) і секцію обробки виключень (необов'язкова). Структура блоку PL / SQL:

**DECLARE** - необов'язково - змінні, константи, курсори

**BEGIN** - обов'язково - команди SQL, керуючі команди PL / SQL

**EXCEPTION** - необов'язково - дії в разі виникнення помилок

**END;** - обов'язково

Блоки PL / SQL складаються з рядків, кожна з яких може містити наступні лексичні елементи: - роздільники (прості і складні); - ідентифікатори (включаючи зарезервовані слова); - літерали; - коментарі.

Ідентифікатори використовуються для іменування програмних модулів, констант, змінних, винятків, курсорів, підпрограм і пакетів. Ідентифікатор може складатися з букв алфавіту цифр (не на першій символічній позиції ідентифікатора), знаків \$ і #, підкреслень. Використання інших символів в позначенні ідентифікатора заборонено. Регістр алфавітних символів не враховується. Деякі з ідентифікаторів, звані зарезервованими словами, мають спеціальний сенс в рамках PL / SQL і не можуть бути перевизначеними.

### 2. Типи даних

Визначені типи даних можна умовно розбити на чотири різновиди:

- 1) скалярні типи (що не мають внутрішніх компонентів);
- 2) складові типи (у яких є внутрішні компоненти, з кожним з яких можна працювати індивідуально);
- 3) посилальні типи;
- 4) LOB-типи, призначені для зберігання об'єктів великого розміру (Large Objects types).

Скалярні типи в свою чергу діляться на чотири різновиди:

- 1) числові типи;
- 2) символічні типи;
- 3) логічний тип (BOOLEAN);
- 4) тип для зберігання часу / дати (DATE).

### 3. Керуючі структури

Для зміни логічного потоку операцій в блоці PL / SQL використовуються наступні керуючі структури:

- конструкції умовного управління з оператором IF;
- простий цикл для безумовного повторного виконання дій;
- цикл FOR для управління кількістю повторів по лічильнику;

- цикл WHILE для управління кількістю повторів по результату перевірки істинності вираження;

- оператор EXIT для виходу з циклу

Приклад:

```
IF v_start > 100 then  
    RETURN (2 * v_start);  
ELSIF v_start >= 50 THEN  
    RETURN (.5 * v_start);  
ELSE  
    RETURN (.1 * v_start);  
END IF;
```

PL / SQL надає кілька шляхів організації циклів для повторного виконання пропозицій.

Простий цикл - це група повторно виконуваних операторів, обмежених операторами LOOP і END LOOP. Кожен раз, як потік управління досягає пропозиції END LOOP, управління передається LOOP. Щоб цикл зробити кінцевим, додають оператор EXIT.

Приклад:

```
LOOP  
    INSERT INTO s_item (ord_id,item_id)  
    VALUES (v_ord_id, v_counter);  
    v_counter:=v_counter+1;  
    EXIT WHEN v_counter>10;  
END LOOP;
```

Цикл FOR має таку ж структуру, як вже розглянуті цикли. Але ключовим словом LOOP передує керуючий оператор, який задає кількість повторів циклу.

Приклад:

```
FOR i IN v_lower..v_upper LOOP  
    v_counter:=v_counter+1;  
    v_output:=i;  
END LOOP;
```

Цикл WHILE використовується для повторення послідовності оператора протягом всього часу, поки виконується задана умова.

Приклад:

```
WHILE v_counter<=10 LOOP  
    INSERT INTO s_item (ord_id, item_id)  
    VALUES (v_ord_id, v_counter);  
    v_counter:=v_counter+1;  
END LOOP;
```

### 3. Обробка винятків

Виняток - це змінна в PL / SQL, порушується під час виконання блоку і припиняє виконання дій в тілі блоку. Якщо PL / SQL збуджує виключення, виконання блоку переривається завжди, але ви можете вказати обробник виключення, який виконає деякі заключні дії. Методи збудження виключення:

- Автоматично в разі виникнення помилки Oracle.

- Явно за допомогою пропозиції RAISE.

Синтаксис опису винятків:

```
EXCEPTION  
    WHEN exception1
```

```
[OR exception2 ...]
THEN
operator1;
operator2;
...
WHEN exception3
[OR exception4 ...]
THEN
operator1;
operator2;
...
WHEN OTHERS
THEN
operator1;
operator2;
...]
```

Для детального ознайомлення з PL/SQL рекомендуємо використати літературу за наступними посиланнями:

1. Довідник по PL/SQL - [Електронний ресурс], режим доступу:  
<https://drive.google.com/file/d/0BzApjGgVUsviSjJLMHpTUWRSdUU/view>
2. Основи мови PL/SQL - [Електронний ресурс], режим доступу:  
<http://www.ict.edu.ru/ft/005965/m08-97.pdf>

## Лабораторна робота № 4. Основні оператори PL/SQL. Приклади реалізації бізнес-логіки

Для відображення на сторінці apex html формату необхідно:

1) Правильно написати html код, який потрібно перевірити (зробити це можна на сторінці <https://codepen.io/pen/>)

2) Для перевірки відображення в Apex, необхідно перейти на вкладку SQL Workshop->SQL Commands, приклад даної вкладки наведено на рисунку 1.

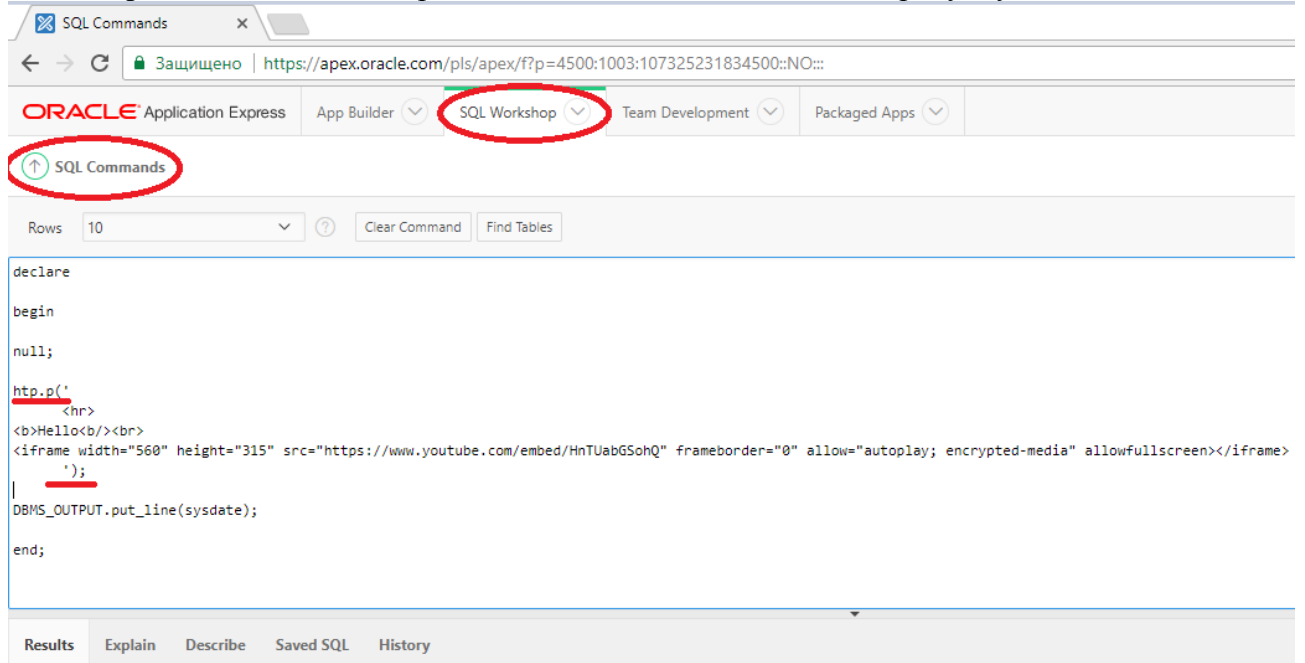


Рисунок 1 – Вкладка SQL Commands

3) Для коректного відображення на сторінці, перед та після html коду необхідно дописати наступне:

```
declare
begin
null;
htp.p(' ');
end;
```

де в htp.p(' ') необхідно вставити потрібний html код. Для відображення дати, необхідно написати DBMS\_OUTPUT.put\_line(sysdate), як це наведено на рисунку 1.

4) Для вставки html безпосередньо на сторінку, необхідно в App Builder вибрати додаток, на який ви хочете додати html, перейти на його редагування, в Rendering, в Content Body, створити новий регіон(клацнути правую кнопку мишки->Create Region). В Region, що знаходиться в правій частині екрану, в Identification вибрати Type, та обрати PL/SQL Dynamic Content. У Source в PL/SQL Code необхідно вставити свій html код та зберегти його, приклад наведено на рисунках 2, 3.

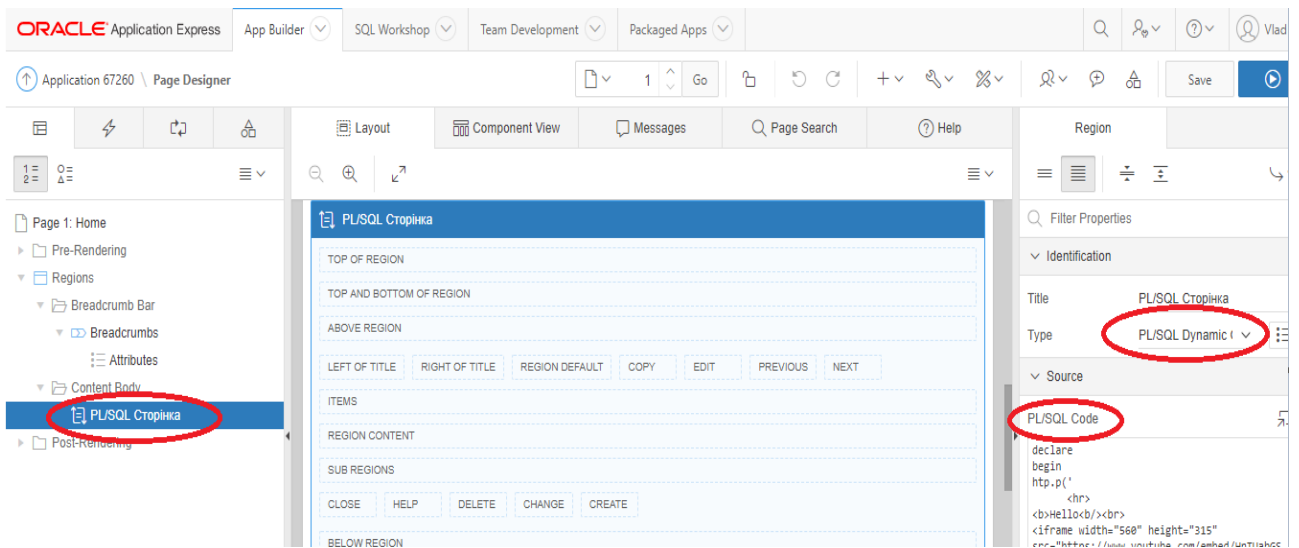


Рисунок 2 – Вставка html коду на сторінку Apex

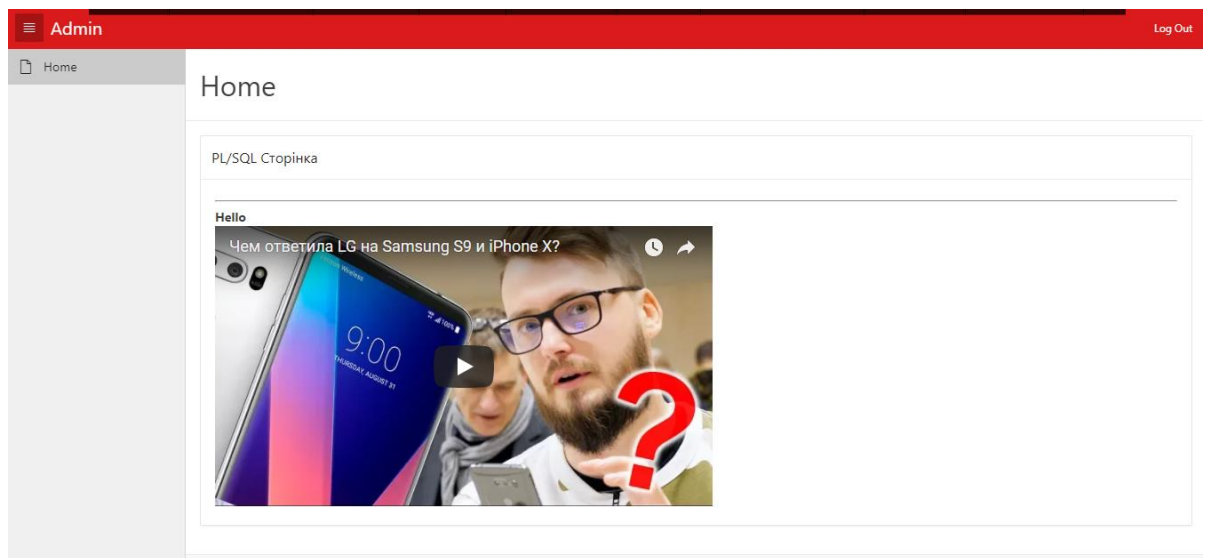


Рисунок 3 – Вигляд apex сторінки

Також html код можна вставити по-іншому. Для цього необхідно створити новий регіон, в Content Body, та вставити в Region->Source->Text html код, вже без додаткового синтаксису, приклад наведено на рисунках 4,5.

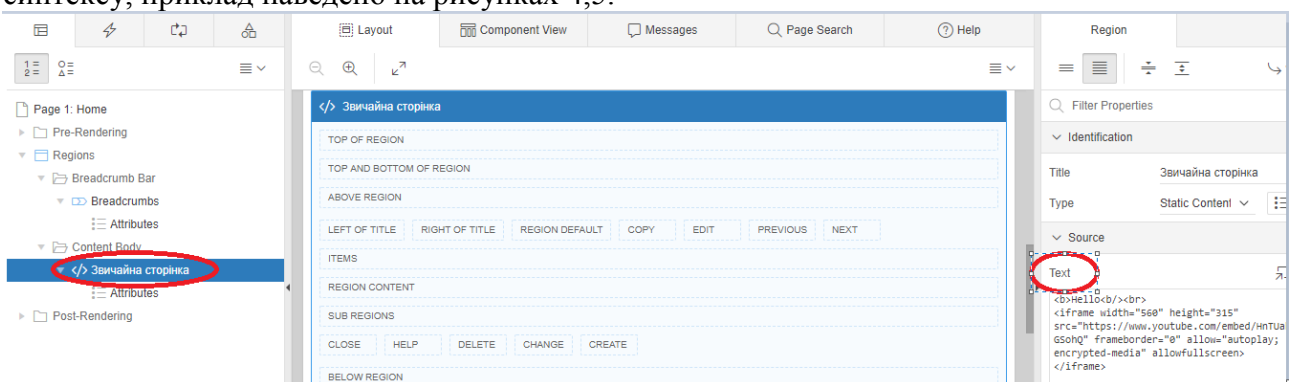


Рисунок 4 – Вставка html коду на сторінку Apex

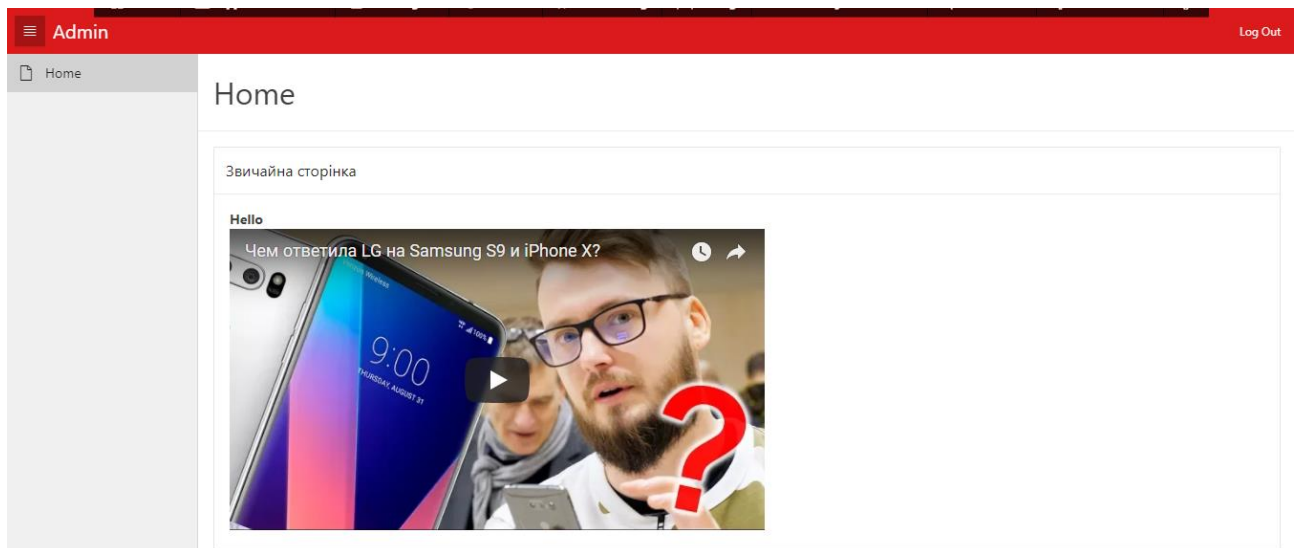


Рисунок 5 –Вигляд арх сторінки

Відмінністю між даними способами є те, що перший використовується, для виводу не тільки html коду, а й для виводу даних з БД, а також під час першого способу перевіряється правильність написаного коду, через .

2. Для підключення CSS та JS стилів до арх сторінки, необхідно:

- 1) Перевірити правильність CSS та JS коду, зробити це можна також на сторінці <https://codepen.io/pen/>.
- 2) В Арех, в App Builder вибрати додаток, до якого хочете підключити CSS, перейти до його редагування та вибрати сторінку. Далі натиснути на title сторінки та в Page, що знаходиться в правій частині екрану знайти CSS->Inline, та вставити свій CSS код. Html необхідно підключати згідно першого уроку, приклад наведено на рисунках 6-7.
- 3) В Арех, в App Builder вибрати додаток, до якого хочете підключити JS, перейти до його редагування та вибрати сторінку. Далі натиснути на title сторінки та в Page, що знаходиться в правій частині екрану знайти JavaScript->Function and Global Variable Declaration, та вставити свій JS код. Html необхідно підключати згідно першого уроку, приклад наведено на рисунках 6-7.

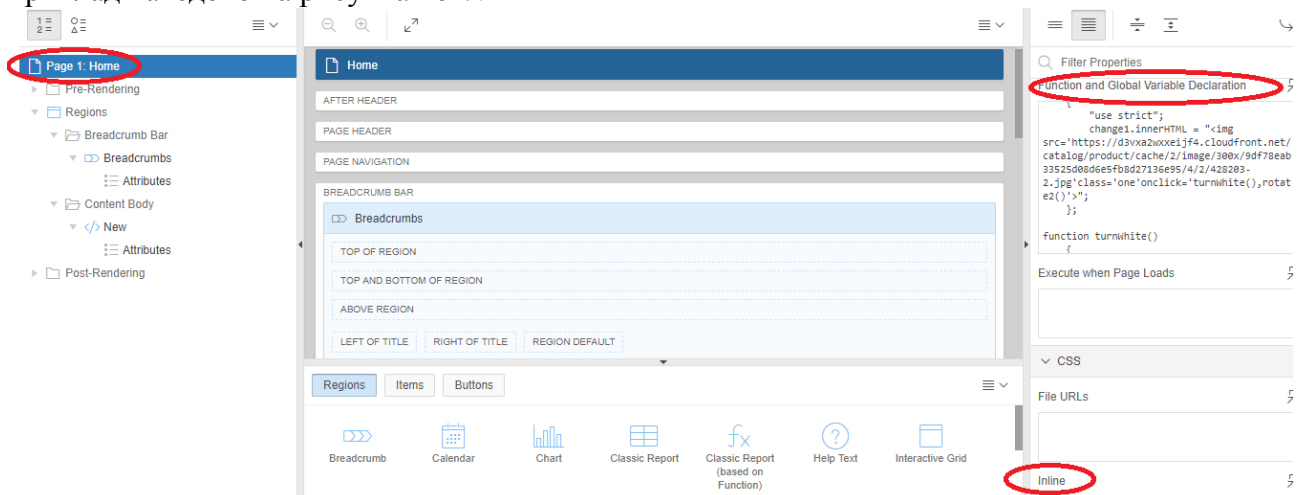


Рисунок 6 – Підключення CSS та JS в арх

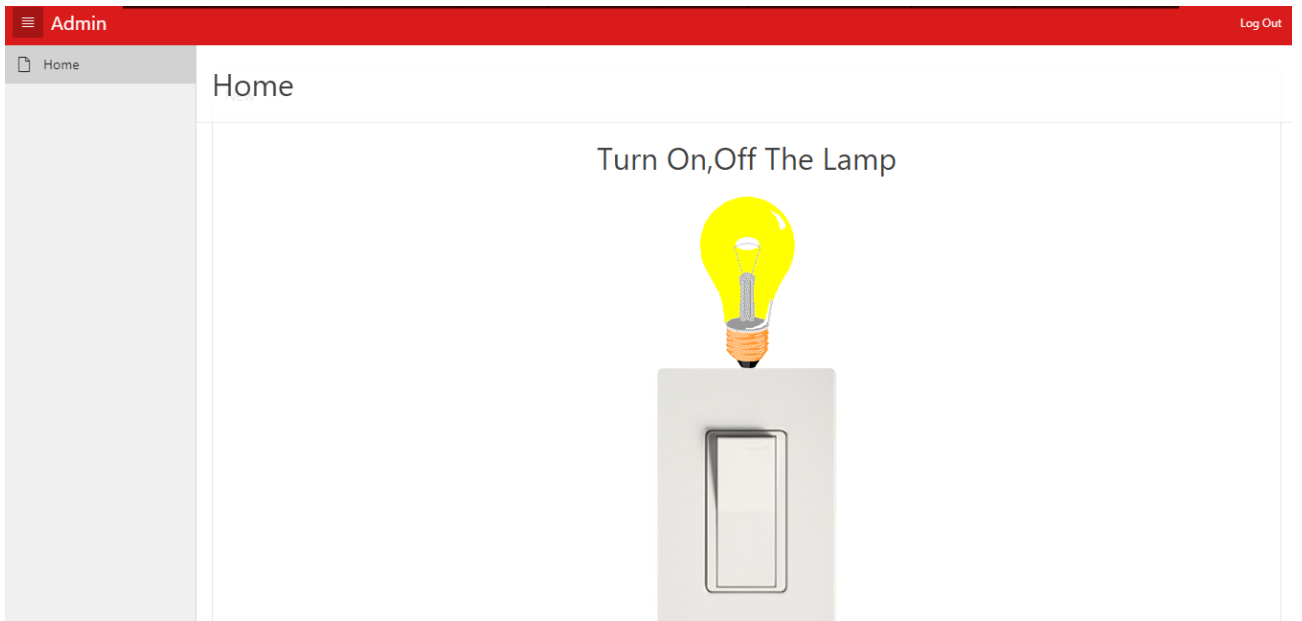


Рисунок 7 – Результат підключення CSS та JS в арх



## Лабораторна робота № 5. Основні оператори PL/SQL. Робота з SQL Developer.

1. В кожній таблиці, в арех є вкладка UI Defaults, яка знаходиться в SQL Workshop->Object Browser->будь-яка таблиця(наприклад Users), де описуються зовнішній вигляд по замовчуванням для компонентів. Тут можна редагувати кожне поле, для його зміни, натиснувши на Edit та вибравши колонку, яку ви хочете відредагувати, приклад наведено на рисунках 1, 2.

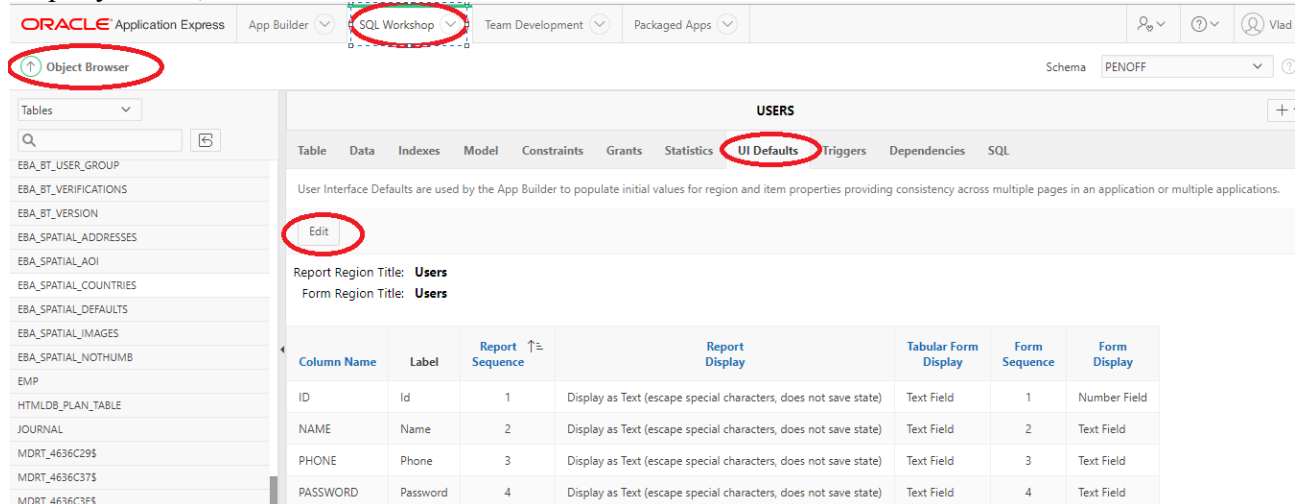


Рисунок 1 – Знаходження UI Defaults

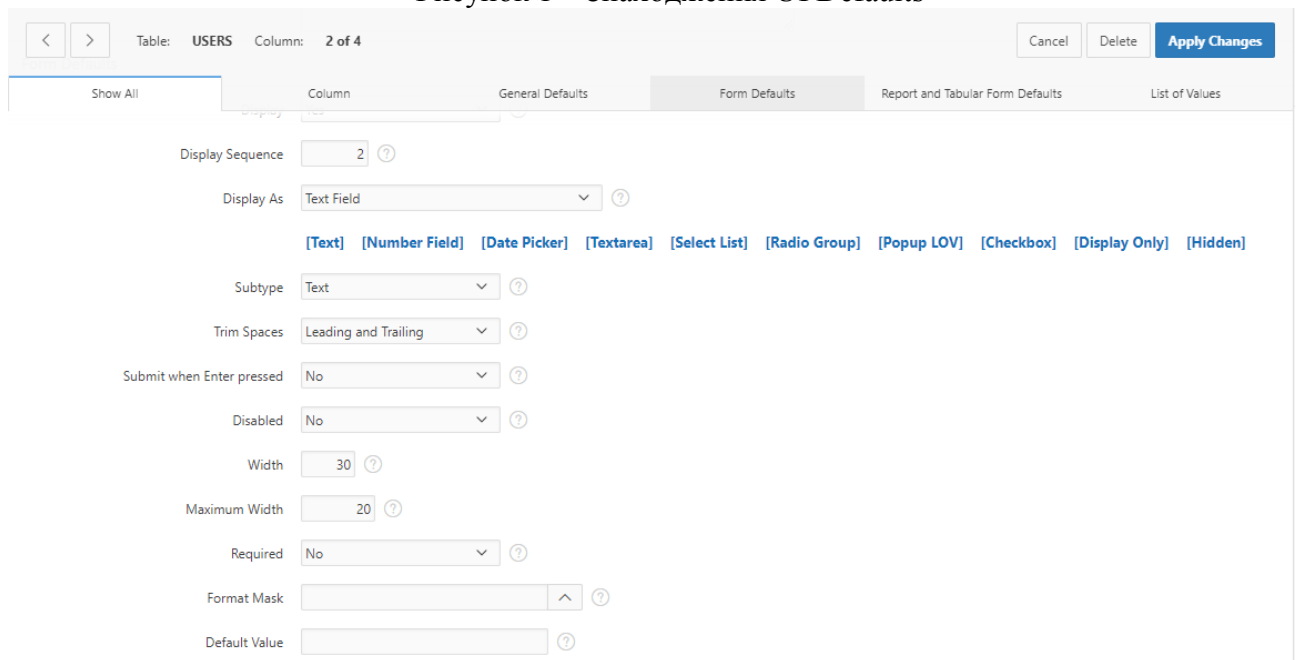


Рисунок 2 - Вкладка редагування name поля таблиці USERS

2. Для видалення сторінки необхідно перейти App Builder, вибрати додаток, в якому зберігається непотрібна сторінка та натиснути на неї. Натиснути на Utilities->Delete page та підтвердити видалення, приклад наведено на рисунку 3.

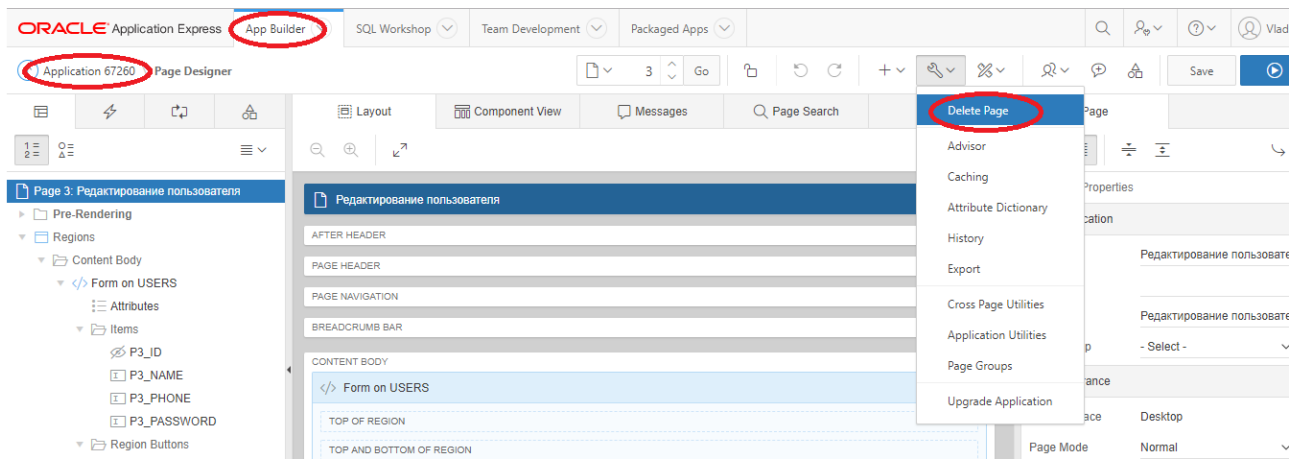


Рисунок 3 – Приклад видалення сторінки

3. Створимо довідник в додатку Admin, для цього необхідно перейти на App Build->Admin та натиснути на Create Page, після чого відкриється вікно мастера налаштувань, де необхідно вибирати наступні вкладки Form->Report with Form on Table. Вам відкриється форма створення. У Вкладці Page Attributes необхідно дотримуватись рисунків 4-7 (якщо номери сторінок, при формуванні форми вже будуть зайняті, можна вибрати довільно).

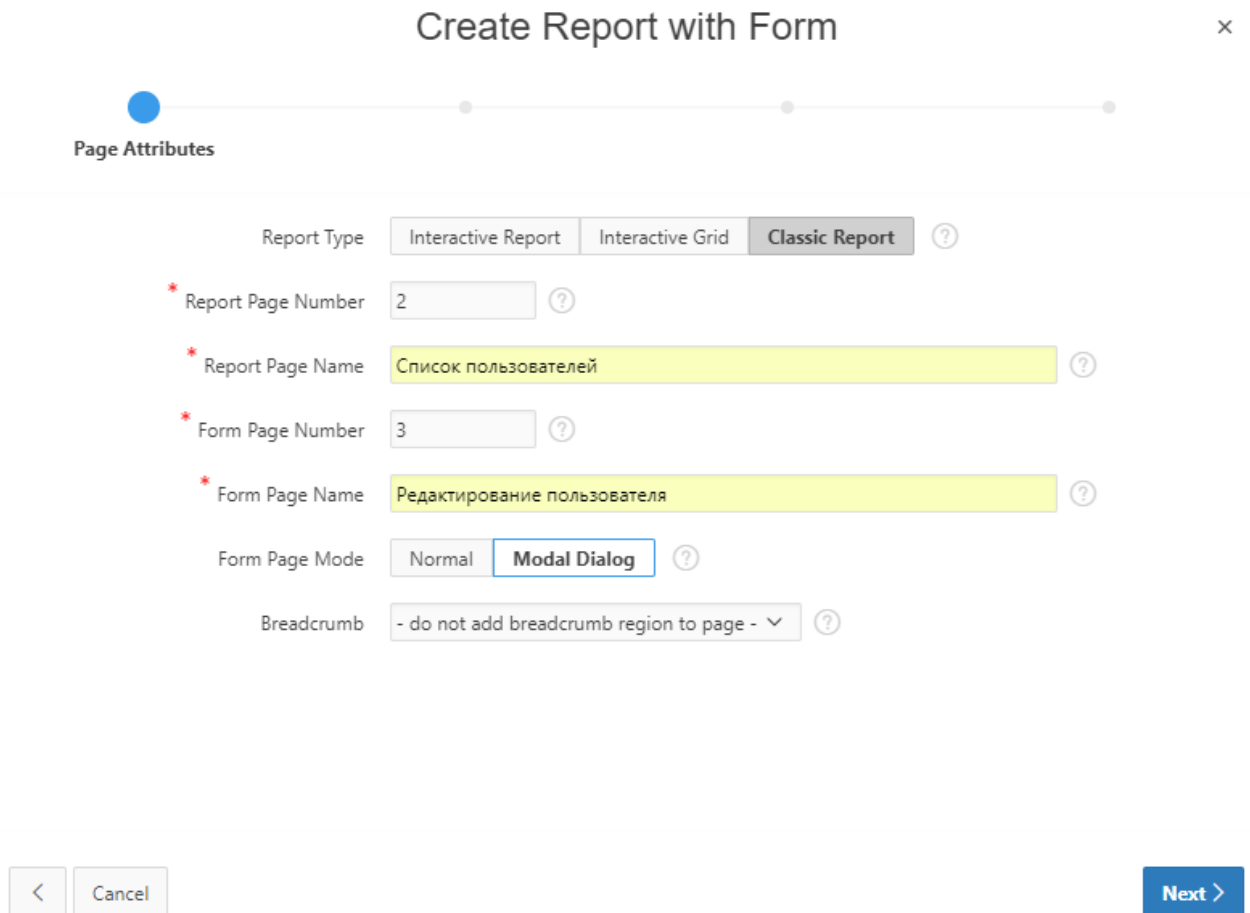


Рисунок 4 – Page Attributes

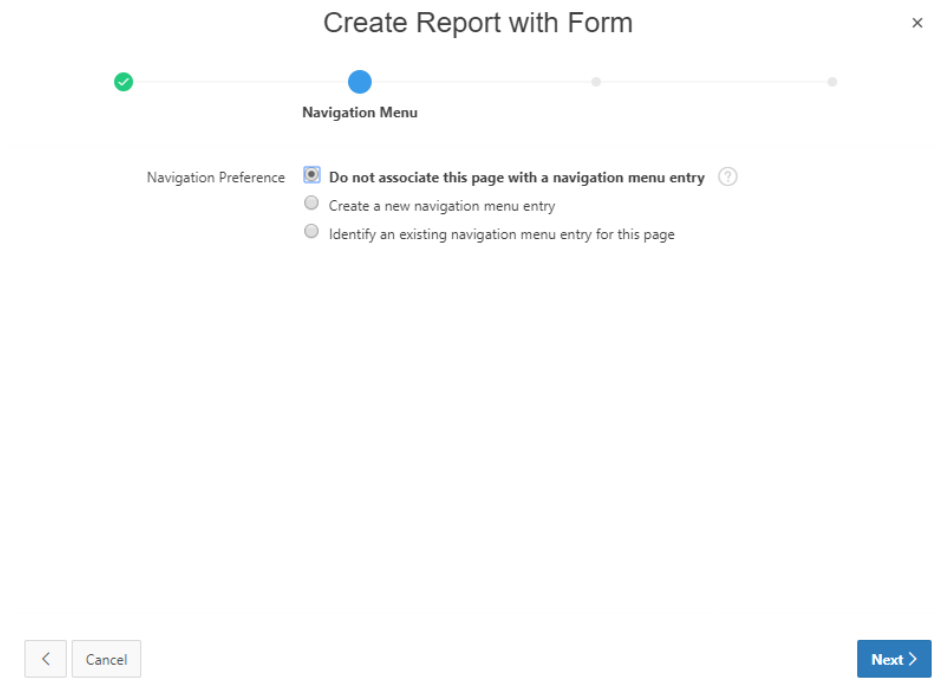


Рисунок 5 – Navigation Menu

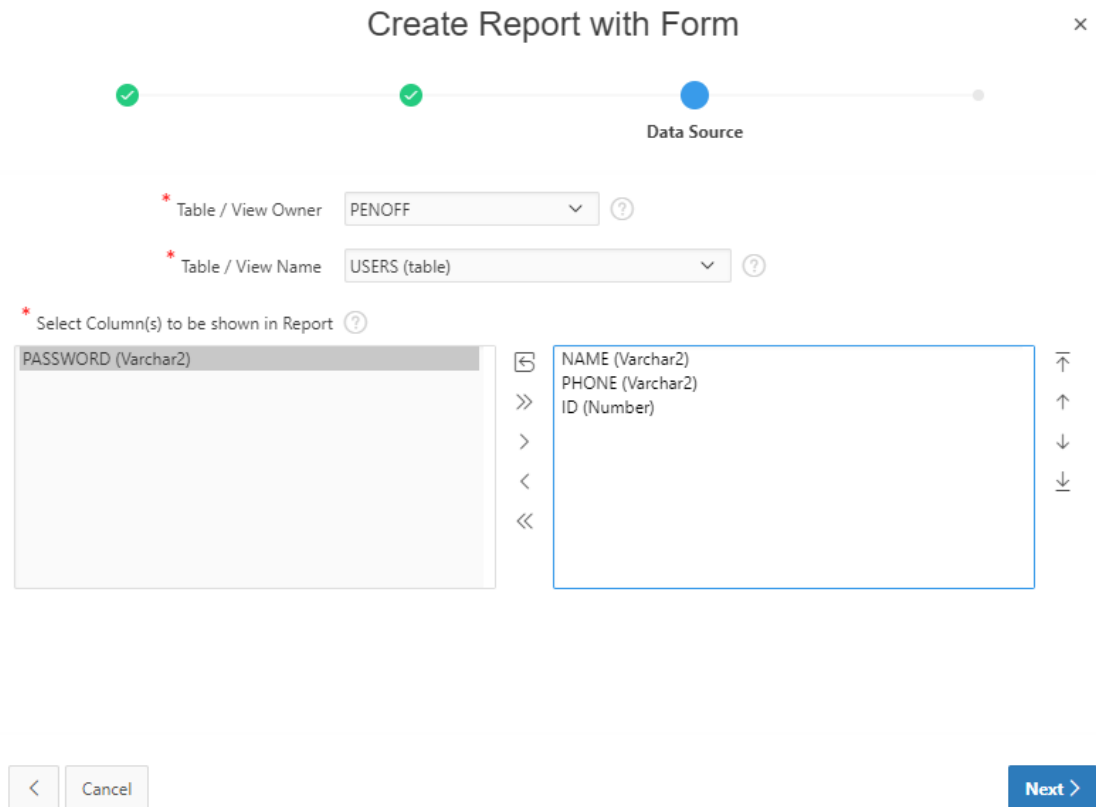


Рисунок 6 – Data Source

### Create Report with Form

×

✓ ✓ ✓ ●

Form Page

\* Primary Key Type:  Managed by Database (ROWID) ?  
 Select Primary Key Column(s)

\* Select Column(s) to be included in Form ?

ID (Number)

NAME (Varchar2)  
PHONE (Varchar2)  
PASSWORD (Varchar2)

\* Primary Key Column 1:  ?  
 Primary Key Column 2:  ?

\* Source for Primary Key Column 1:  Existing Trigger ?  
 Existing Sequence

< Cancel
Create

Рисунок 7 – Form Page

4. Створимо довідник товарів. Для цього необхідно додати нову таблицю, необхідно перейти в SQL Workshop->Object Browse, натиснути на +, що знаходиться в правій частині екрану та вибрати table. Після чого вам відкриється майстер налаштування таблиці, де можна написати назву таблиці, назвати колонки, вибрати їх тип та довжину, детально процес створення таблиці наведено на рисунках 8-12.

● ○ ○ ○ ○

Columns

\* Table Name:  ?

Preserve Case

Column Name	Type	Precision	Scale	Not Null	Identity	Move
<input type="text" value="id"/>	NUMBER			<input type="checkbox"/>	- None -	^ v
<input type="text" value="name"/>	VARCHAR2		<input type="text" value="199"/>	<input type="checkbox"/>		^ v
<input type="text"/>	- Select Datatype -					^ v
<input type="text"/>	- Select Datatype -					^ v
<input type="text"/>	- Select Datatype -					^ v
<input type="text"/>	- Select Datatype -					^ v
<input type="text"/>	- Select Datatype -					^ v
<input type="text"/>	- Select Datatype -					^ v

Cancel
Next >

Рисунок 8 – Columns

На вкладці Primary Key можливо вибрати первинний ключ, з наявних варіантів:

- No Primary Key - не створювати первинний ключ передбачає самостійне формування поля id.
- Populated from a new sequence - створити первинний ключ за допомогою послідовності (автоматична послідовність).

- Populated from an existing sequence - створити первинний ключ за допомогою послідовності, яка вже є.

**Primary Key**

Table name:

Primary Key:  No Primary Key  **Populated from a new sequence**  Populated from an existing sequence  Not populated  Populated by Identity column

\* Primary Key Constraint Name:

\* Primary Key:

\* Sequence Name:

**Primary Key**  
A primary key allows each row in a table to be uniquely identified.

If you select to populate your primary key from a new sequence, you will be prompted to enter the new sequence's name. If you select to populate your primary key from an existing sequence, you will be prompted to select the sequence. Both these methods result in the generation of a trigger against your table. You can also select to not populate your primary. This is the only method that allows you to define a composite primary key made up of more than two columns.

Cancel Next >

Рисунок 9 – Primary Key

**Foreign Keys**

Foreign Key	Columns	Referenced Table	Referenced Columns	Action
-------------	---------	------------------	--------------------	--------

**Add Foreign Key**

Name:

**Disallow Delete**  
 Cascade Delete  
 Set Null on Delete

Select Key Column(s):

References Table:

Referenced Column(s):

**Add**

Рисунок 10 – Foreign Key

В Constraints можна перевірити дані на унікальність, для цього необхідно натиснути на прапорець Unique та вибрати колонку в Key Column(s), яка повинна бути унікальною та

натиснути на Add, для додавання після чого колонка буде відображатися зліва вверху, приклад наведено на рисунку 11.

Constraint Name	Type	Column(s)/Check
PRODUCT_GRP_NAME	Unique	NAME X

Constraint Type  Check  Unique

Check Condition

Key Column(s) ID NAME

Рисунок 11 – Costraints

В Confirm, все те, що ми вибирали в майтері налаштувань можна подивитися в мові SQL, для цього необхідно натиснути на SQL, приклад наведено на рисунку 12

Please confirm your request.

Schema: PENOFF

Table name: PRODUCT\_GRP

SQL

```
CREATE table "PRODUCT_GRP" (  
  "ID"          NUMBER,  
  "NAME"       VARCHAR2(199),  
  constraint "PRODUCT_GRP_PK" primary key ("ID")  
)  
/  
  
CREATE sequence "PRODUCT_GRP_SEQ"  
/  
  
CREATE trigger "BI_PRODUCT_GRP"  
  before insert on "PRODUCT_GRP"  
  for each row  
  begin  
    if :NEW."ID" is null then
```

Рисунок 12 – Confirm

Аналогічно створимо таблицю товарів, що буде підпорядковуватися попередній таблиці, згідно рисункам 13-16

Columns

\* Table Name  ?

Preserve Case

Column Name	Type	Precision	Scale	Not Null	Identity	Move
<input type="text" value="id"/>	NUMBER	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	- None -	^ v
<input type="text" value="name"/>	VARCHAR2		199	<input type="checkbox"/>		^ v
<input type="text" value="product_grp"/>	NUMBER	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	- None -	^ v
<input type="text" value="price"/>	NUMBER	19	2	<input type="checkbox"/>	- None -	^ v
<input type="text" value="cost"/>	NUMBER	19	2	<input type="checkbox"/>	- None -	^ v
<input type="text" value="descr"/>	VARCHAR2		2000	<input type="checkbox"/>		^ v
<input type="text" value="url"/>	VARCHAR2		199	<input type="checkbox"/>		^ v
<input type="text"/>	- Select Datatype -					^ v

Add Column

Рисунок 13 – Columns

## Create Table

Primary Key

Table name: **PRODUCT** ?

Primary Key:  No Primary Key ?  
 **Populated from a new sequence**  
 Populated from an existing sequence  
 Not populated  
 Populated by Identity column

\* Primary Key Constraint Name  ?

\* Primary Key  ?

\* Sequence Name  ?

### Primary Key

A primary key allows each row in a table to be uniquely identified.

If you select to populate your primary key from a new sequence, you will be prompted to enter the new sequence's name. If you select to populate your primary key from an existing sequence, you will be prompted to select the sequence. Both these methods result in the generation of a trigger against your table. You can also select to not populate your primary. This is the only method that allows you to define a composite primary key made up of more than two columns.

Рисунок 14 – Primary Key

Вкладка Foreign Key підказує БД, як таблиці між собою пов'язані. За допомогою неї перевіряється:

- Disallow Delete - випадкове видалення,
  - Cascade Delete - якщо видаляється група, в якій є товари, то автоматично будуть видалятися товари.
  - Set Null on Delete – якщо видаляється група, то замість неї буде ставитись 0
- В вкладці Select Key Column(s) перевіряється, яка саме колонка повинна перевірятись

References Table – вибирається таблиця, з якою ви хочете зв'язати поточну  
 Referenced Column(s) – вибирається колонка, з якою буде зв'язуватися таблиця  
 Після вибору зв'язаної таблиці необхідно натиснути на Add, що знаходиться в правій частині екрану, приклад наведено на рисунку 15.

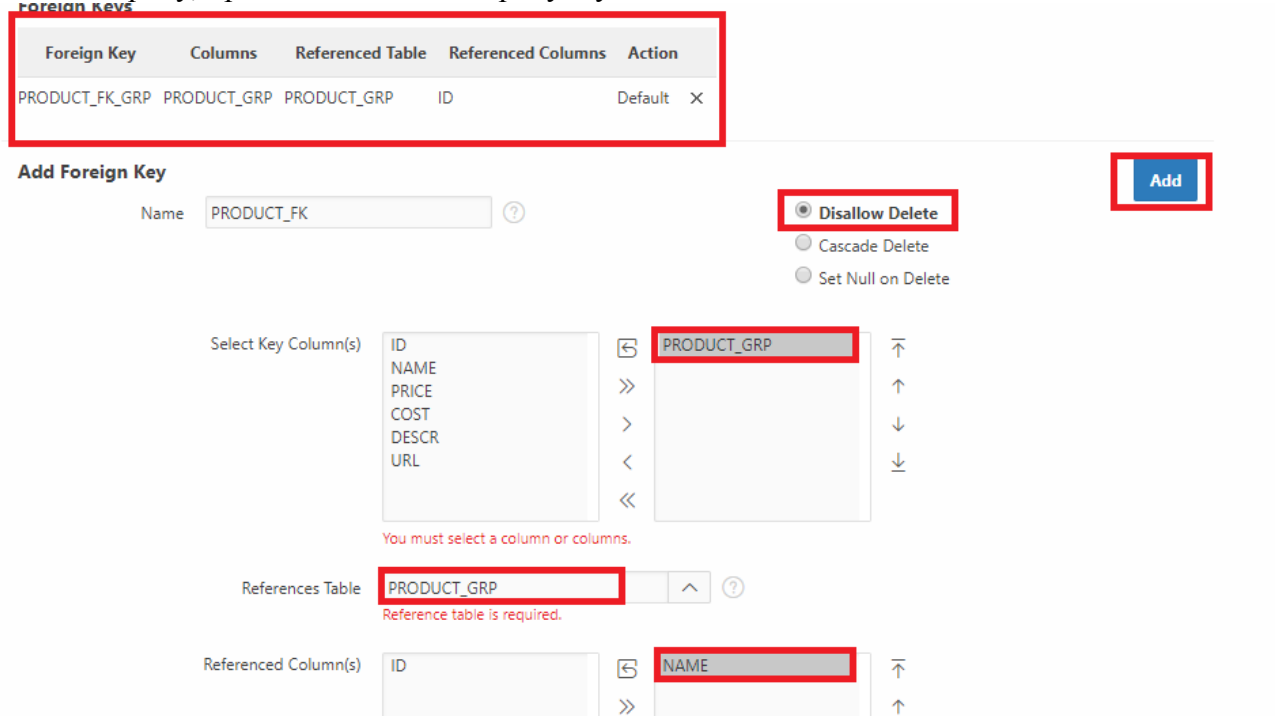


Рисунок 15 - Foreign Key

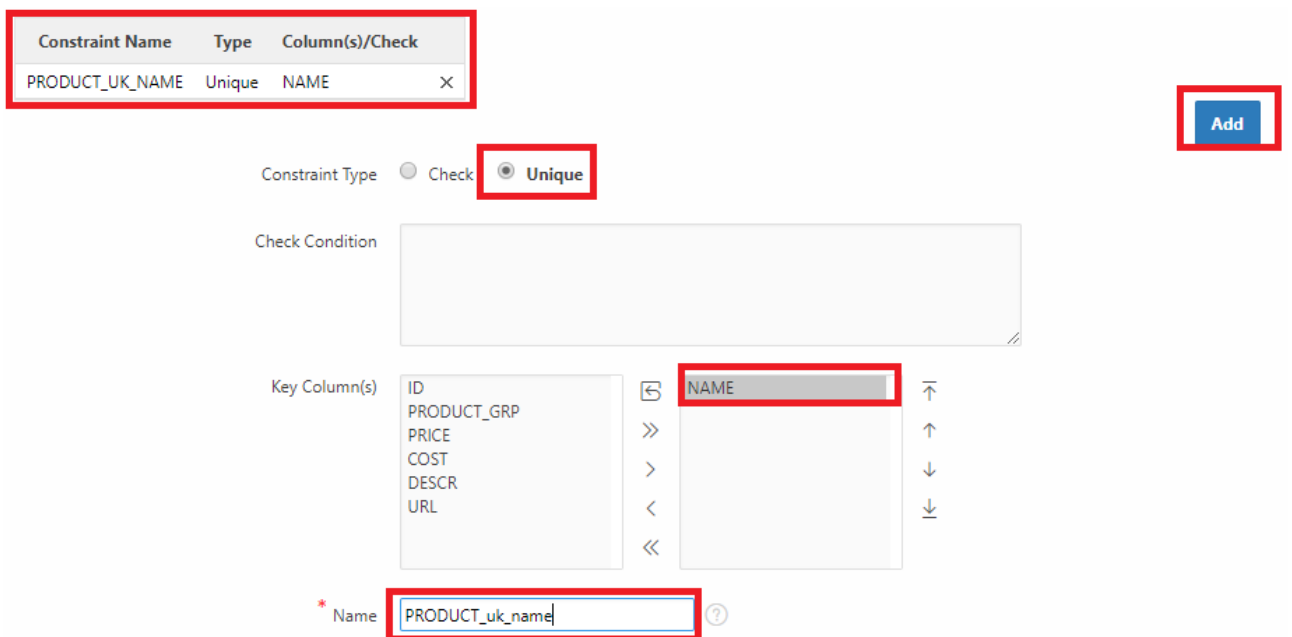


Рисунок 16 - Constraints

Вибрати дані пункти також можна без майстера налаштувань, для цього необхідно перейти на SQL Workshop->Object Browser->вибрати таблицю->Constraint->Create, приклад наведено на рисунку 17



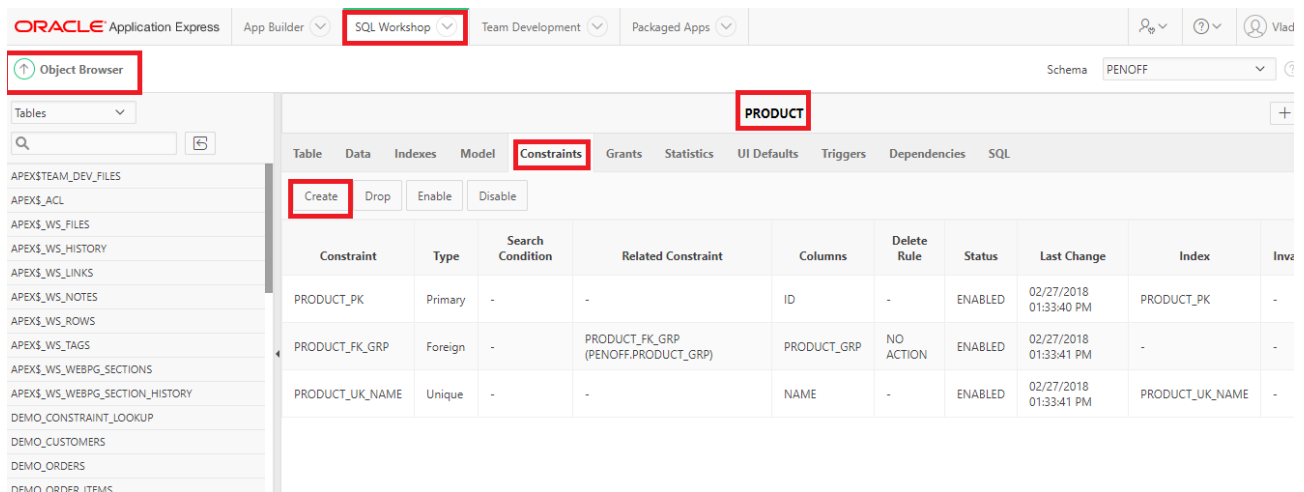


Рисунок 17 – Налаштування таблиці

Для відображення залежності таблиці необхідно перейти на вкладку Model, яка знаходиться в SQL Workshop->Object Browser->вибрати таблицю->Model, приклад наведено на рисунку 18

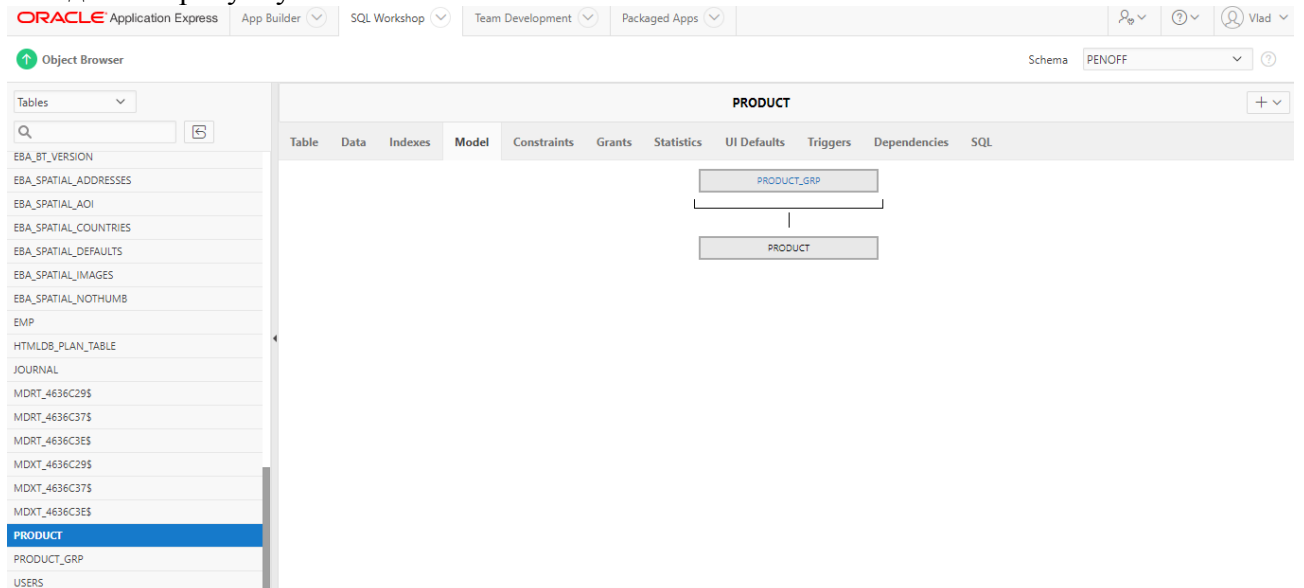


Рисунок 18 – Залежність таблиці

5. Створимо редагування груп товарів, для цього необхідно перейти на App Builder->вибрати додаток->Create Page->Form->Report with Form on Table. В формі створення необхідно слідувати рисункам 19-22.

## Create Report with Form ×

● Page Attributes

Report Type: Interactive Report | Interactive Grid | **Classic Report** ?

\* Report Page Number: 3 ?

\* Report Page Name: Группы товаров ?

\* Form Page Number: 4 ?

\* Form Page Name: Редактирование группы ?

Form Page Mode: Normal | **Modal Dialog** ?

Breadcrumb: - do not add breadcrumb region to page - ?

Рисунок 19 – Page Attributes

✔ ● Navigation Menu

Navigation Preference:  Do not associate this page with a navigation menu entry ?  
 **Create a new navigation menu entry**  
 Identify an existing navigation menu entry for this page

\* New Navigation Menu Entry: Группы товаров ?

Parent Navigation Menu Entry:
 

- No parent selected - ?
- Home
- Регистрация

Рисунок 20 – Navigation Menu

✔ ✔ ● Data Source

\* Table / View Owner: PENOFF ?

\* Table / View Name: PRODUCT\_GRP (table) ?

\* Select Column(s) to be shown in Report ?

	<input type="checkbox"/> ID (Number) <input type="checkbox"/> NAME (Varchar2)	⇅ ↑ ↓ ⇅
--	--	------------------

Рисунок 21 –Data Source

Form Page

\* Primary Key Type:  Managed by Database (ROWID)  Select Primary Key Column(s)

\* Select Column(s) to be included in Form

Name	Type
NAME	Varchar2
ID	Number

\* Primary Key Column 1: ID (Number)

Primary Key Column 2: - Select Column -

\* Source for Primary Key Column 1:  Existing Trigger  Existing Sequence

Рисунок 22 – Form Page

Далі спробуємо додати товари в групу, приклад наведено на рисунку 23.



Рисунок 23 – Додані товари

Аналогічно створимо список та редагування товарів, створення даної сторінки в майстрі налаштувань ідентичне створенню сторінки групи товарів, окрім першої вкладки, яка наведена на рисунках 24-25.

**Page Attributes**

Report Type: **Interactive Report** | Interactive Grid | Classic Report ?

\* Report Page Number:  ?

\* Report Page Name:  ?

\* Form Page Number:  ?

\* Form Page Name:  ?

Form Page Mode: **Normal** | Modal Dialog ?

Breadcrumb:  ?

Parent Entry:  ?

\* Entry Name:  ?

Рисунок 24 – Page Attributes

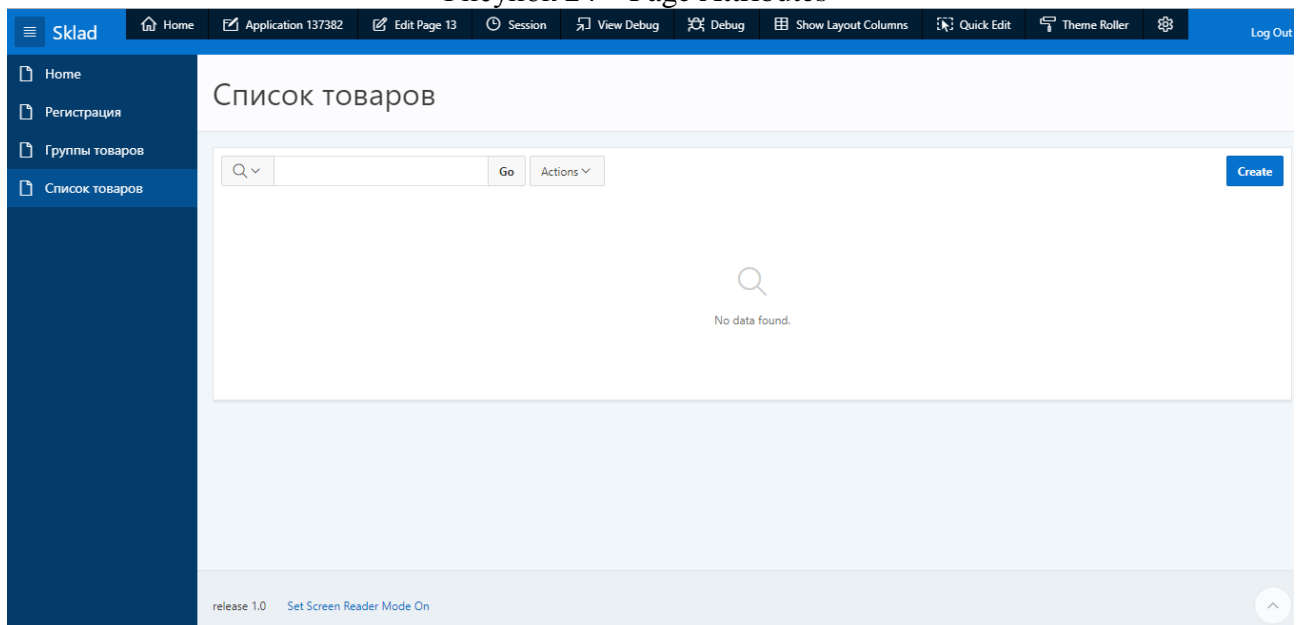


Рисунок 25 – Результат створення сторінки список товарів

Для заповнення форми створення списку товарів нам в колонці Product Grp необхідно зробити випадаюче меню, для цього на сторінці Список товаров, натиснемо на кнопку Create, та у верхньому меню виберемо Edit Page, приклад наведено на рисунку 26.

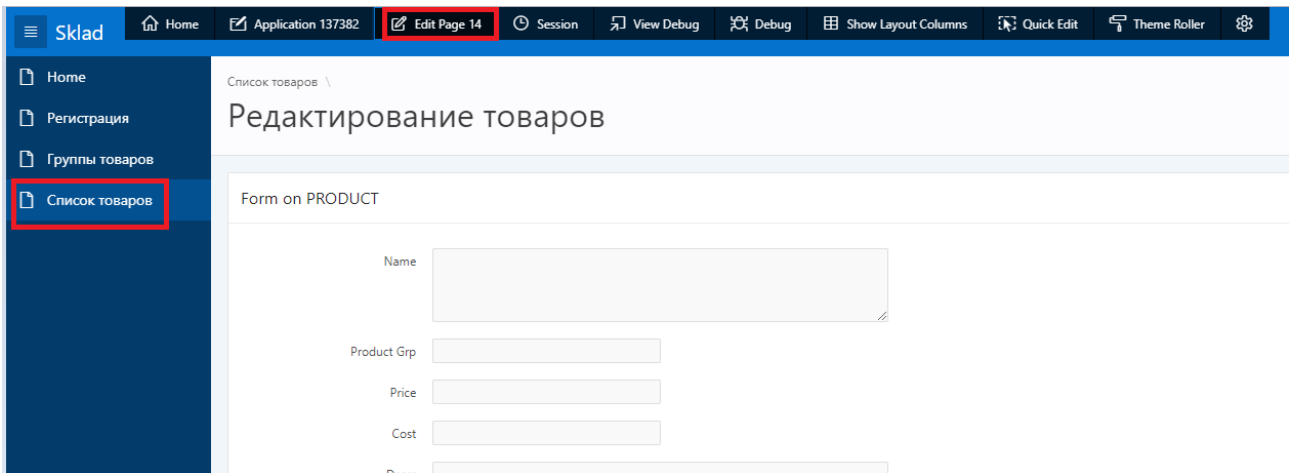


Рисунок 26 – Перехід на редагування сторінки

Далі необхідно в Content Body знайти Items та PRODUCT\_GRP та в Page Item в Identification в Type вибрати Select List. Далі в Page Item в List of Values необхідно вибрати Type – SQL Query та вказати запит, який буде приймати участь в даному запиті:

```
select name,id
from product_grp
order by 1
```

Приклад виконання наведено на рисунках 27-28.

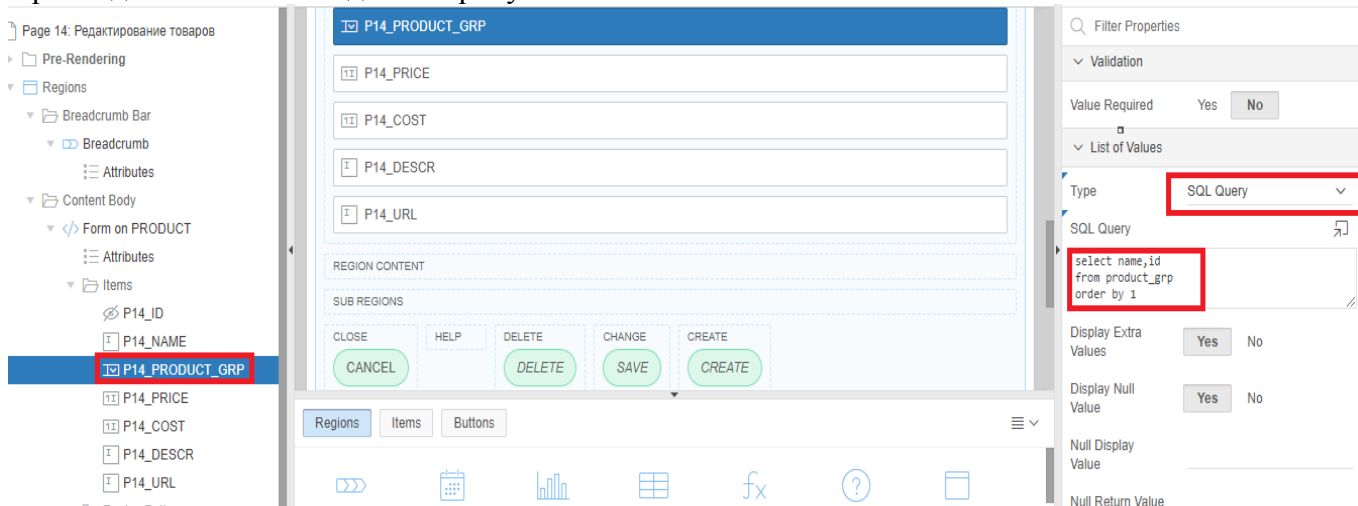


Рисунок 27 –налаштування випадаючого списку

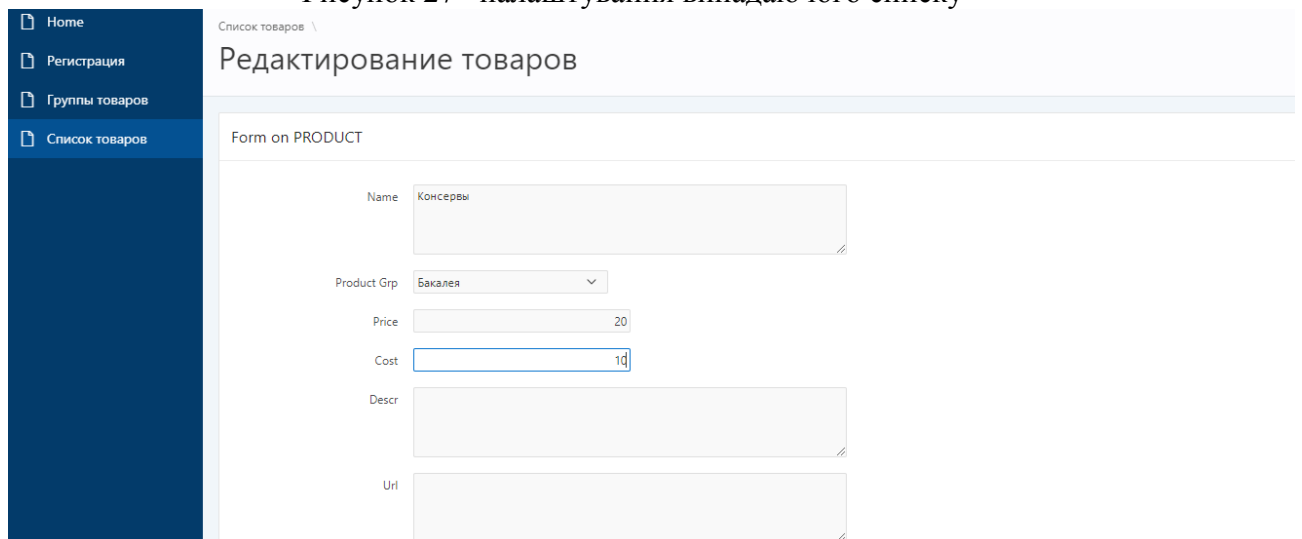


Рисунок 28 – тестування випадаючого списку

Для відображення списку по правій стороні, необхідно вибрати Right Side Column, який знаходиться на сторінці списку товарів в Arrеаgаnсе, приклад наведено на рисунку 29-30.

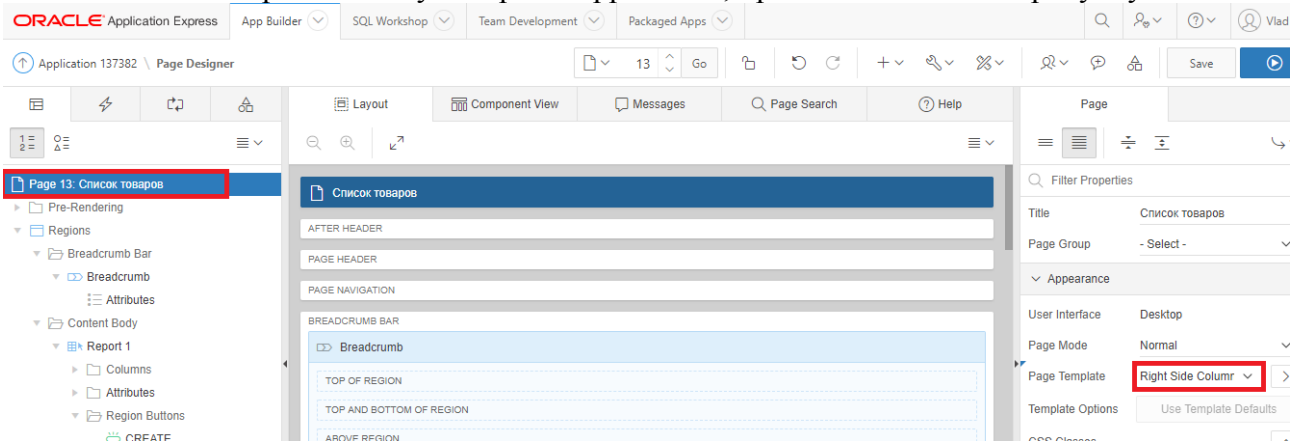


Рисунок 29 – налаштування списку по правій стороні

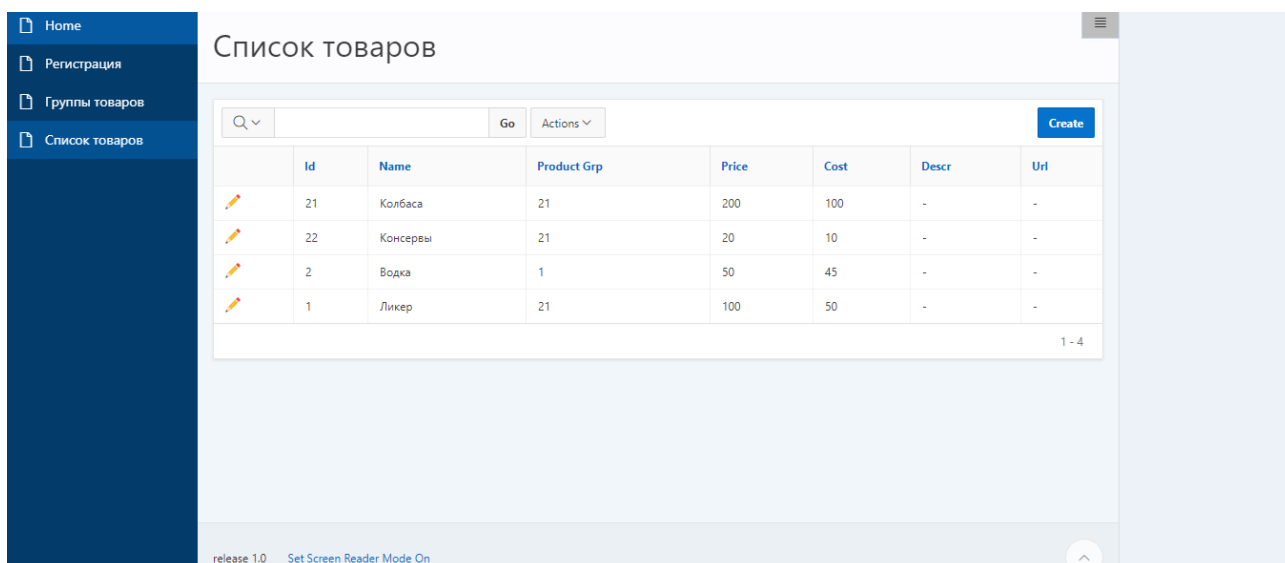


Рисунок 30 – налаштування списку по правій стороні

Створимо новий «регіон» під назвою «Операції», для цього необхідно натиснути на Content Body правою клавішею мишки та створити регіон (Create Region). В Region->Position виберемо Right Column та перенесемо клавішу Create в новий регіон, приклад наведено на рисунках 31-32.

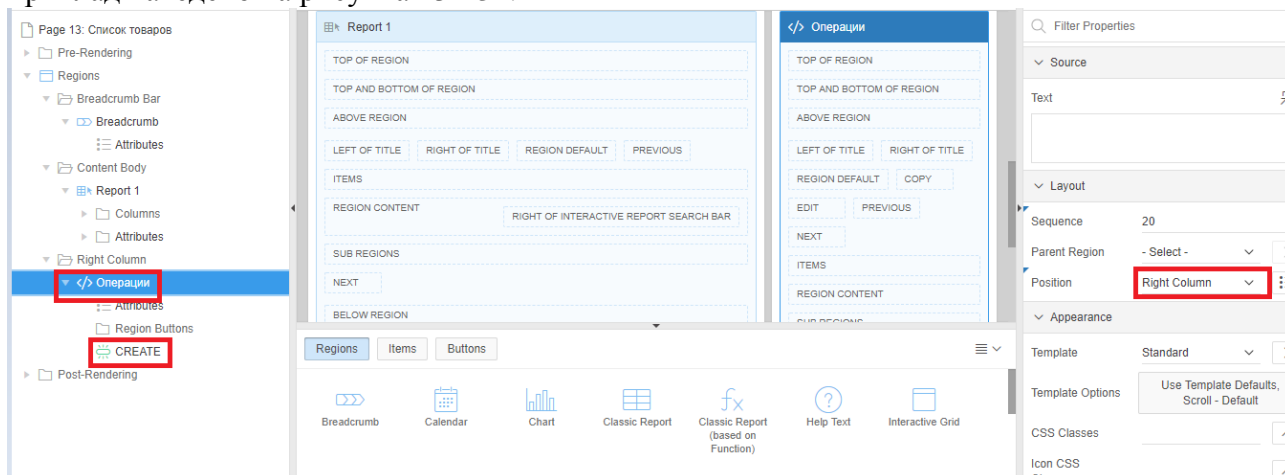


Рисунок 31 – налаштування клавіші Create справа

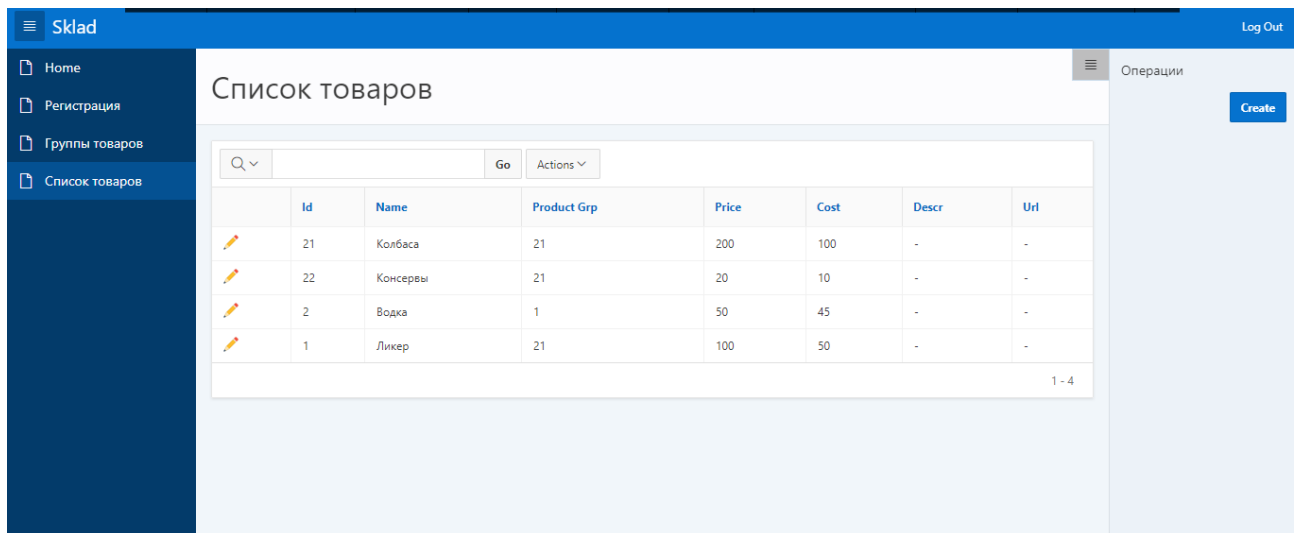


Рисунок 32 – налаштування клавiши Create справа

Створимо новий рiгiон групи, аналогiчно до попереднього, але в Right Column та створимо в данiй групi позицiю(Create Item), перейменувавши її в \_GRP та змiнимо тип в Indefication->Type на Select List та в List of Values->Type виберемо SQL Query та в поле впишемо:

```
select name,id from product_grp
order by 1
```

Для кращого сприймання натиснемо на рiгiон «Группы» та в Region змiнимо Template Option, як наведено на рисунку 33.

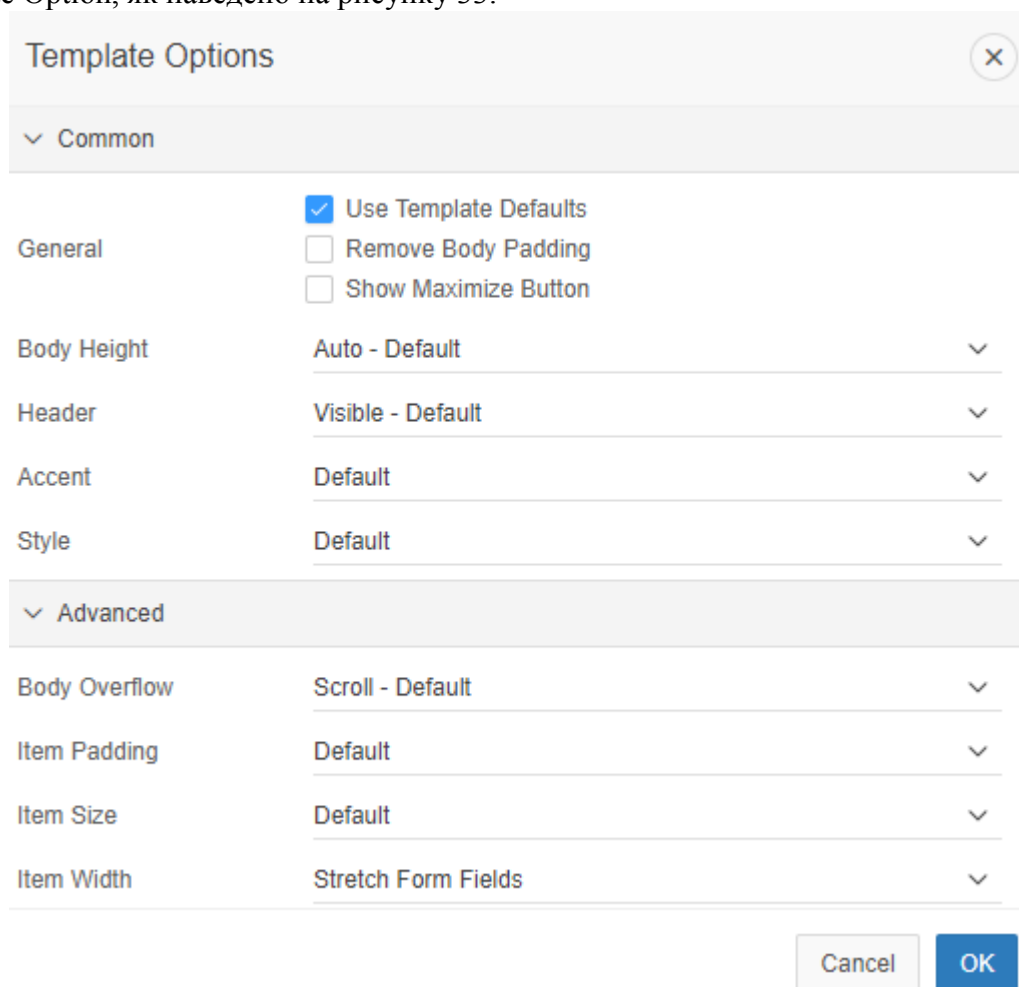


Рисунок 33 – налаштування Template Option

Для редагування саме не сторінці в допоміжній панелі, що за замовчуванням розташована внизу, необхідно вибрати Quick Edit, навести на поле, яке ми хочемо відредагувати та натиснути на значок справа зверху, приклад наведено на рисунку 34.

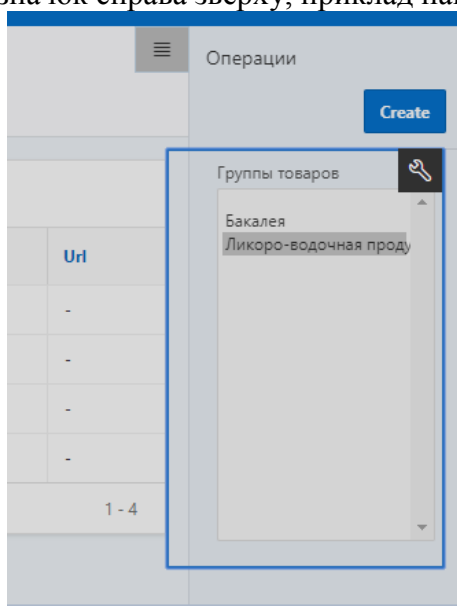


Рисунок 34 – Редагування через Quick Edit

Якщо зайти в Template, для клавiшi Create, можна змiнити її стиль, розмiр, ширину та iншi параметри.

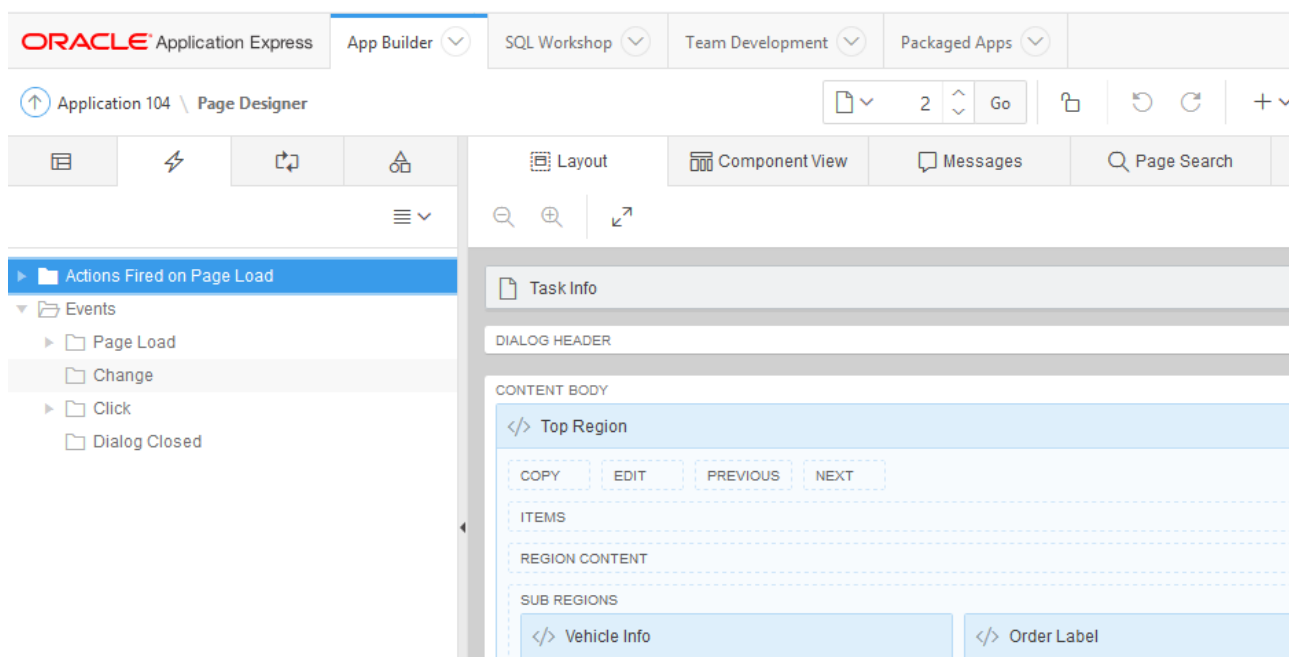
Для редагування полiв таблицi необхідно перейти в Attributes, для Report 1, та в Template змiнити на ваш варiант.

Також можна використати стилi саме на сторiнцi арх, для цього в допомiжнiй панелi, натиснути Theme Roller, де ви можете змiнити тему, а також додати власний CSS



## Лабораторна робота № 6. Динамічні події в Apex. Структурний елемент “дерево”

В Apex динамічними подіями називаються події, які виконуються на стороні клієнта. Це JavaScript та jQuery сховані за інтерфейсом Apex. Доступ до подій можна отримати двома способами. Перший спосіб зображений на рисунку 1, - це перейти на вкладку



“Dynamic Actions” в меню сторінки.

Рисунок 1 — Відкрита вкладка Dynamic Actions

Другий спосіб — це створити подію натиснувши ПКМ на елементі, для якого необхідно додати подію.

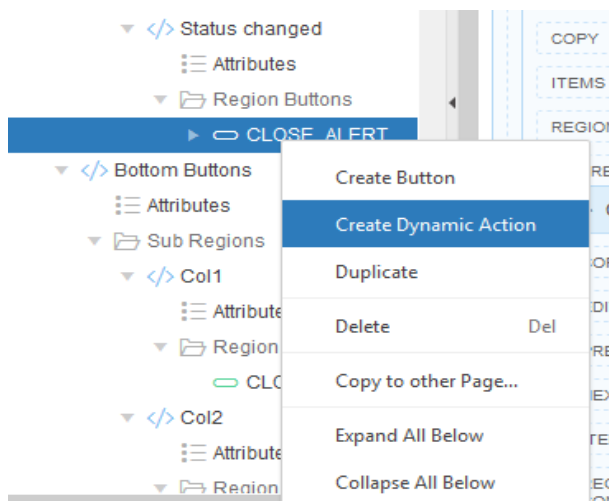


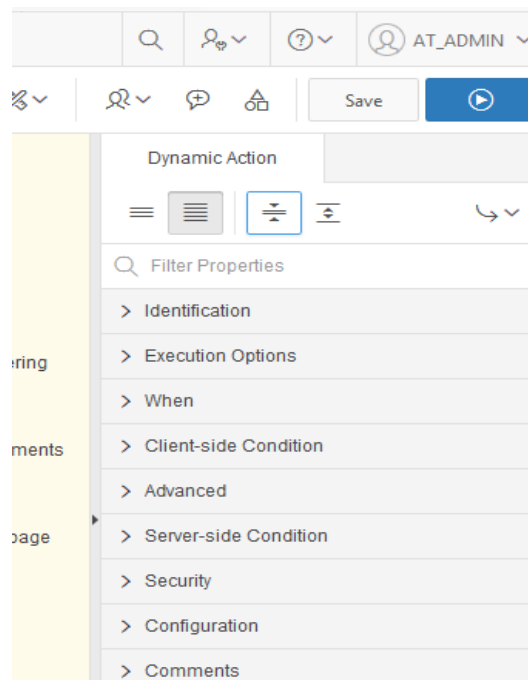
Рисунок 2 — Створення динамічної події чере маню ПКМ.

Обидва способи однакові, і створена подія другим способом одразу з'явиться у вкладці Dynamic Actions. Відмінністю є лише те, що при використанні другого способу в параметри події заноситься об'єк до якого вона прикріплена. При використанні першого способу цю операцію доведеться зробити вручну.

В меню динамічних подій (розміщується справа) є такі пункти меню:

- **Identification** – пункт меню, в якому вказується назва події

- **Execution Options** — пункт, в якому вказується позиція події в черзі на виконання.
- **When** — В полі Event вибирається системна подія, яка буде викликати спрацювання нашої події. Selection type — елемент до якого прикріплюється подія.
- **Client-side Condition** — Умова, яка виконується перед спрацюванням нашої події.
- **Advanced** — В полі Event Score вказується принцип спрацювання події. Static – звичайна подія, яка переривається PPR(Partial Page Refresh); Dynamic — подія, яка не переривається PPR; Once – подія викликається один раз.
- **Server-side Condition** — Умова, яка виконується на сервері
- **Security** — пункт меню для керування доступом
- **Configuration** - Пункт меню призначений для підключення/відключення компоненту (якщо ваша сторінка - компонент)



- **Comments** — Коментар

Рисунок 3 — Меню динамічної події

Після створення динамічної події, в меню Dynamic Actions з'явиться два вкладення true та false. В цих вкладеннях створюється дія яка буде виконуватися під час спрацювання або не спрацювання створеної події. Створити нову true або false дію можна натиснувши ПКМ на відповідному елементі.

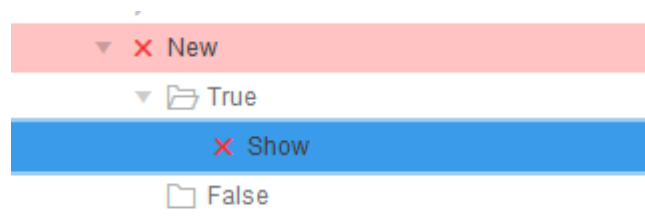


Рисунок 4 — Вигляд створеної події в меню Dynamic Actions

До щойно створеної дії в правому меню є такі пункти:

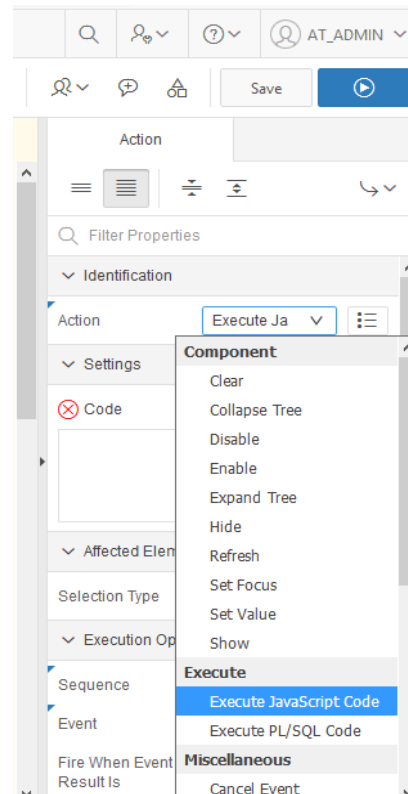
- **Identification** – В полі Action вибирається js подія, яка виконається
- **Settings** — Відображає всі елементи в один рядок (При табличній сітці)
- **Affected Element** — Елемент до якого виконається дія. Може бути: елемент, регіон,

колонка в таблиці, або jQuery селектор.

- **Execution Options** — Вказуються параметри події
- **Comments** — Коментар

Арех підтримує таку технологію як аяx. Для того, щоб відправити дані на сервер без перезавантаження сторінки необхідно в полі Action вибрати Execute PL/SQL Code і нижче ввести PL/SQL запит. Після поля для введення є два параметри: Items to Submit та Items to Return. В першому параметрі необхідно вказати елементи сторінки, які треба передати на сервер. В другому полі вказуються елементи, значення яких необхідно повернути з сервера.

У випадку, якщо вбудованих функцій недостатньо для реалізації якогось складного



ефекту, в полі Action є Execute JavaScript Code, де можна ввести свій власний код.

Рисунок 5 — Дій для виконання PL/SQL та JavaScript кода

### Структурний елемент “дерево”

В загальному випадку дерево виглядає так, як представлено на рисунку 6.

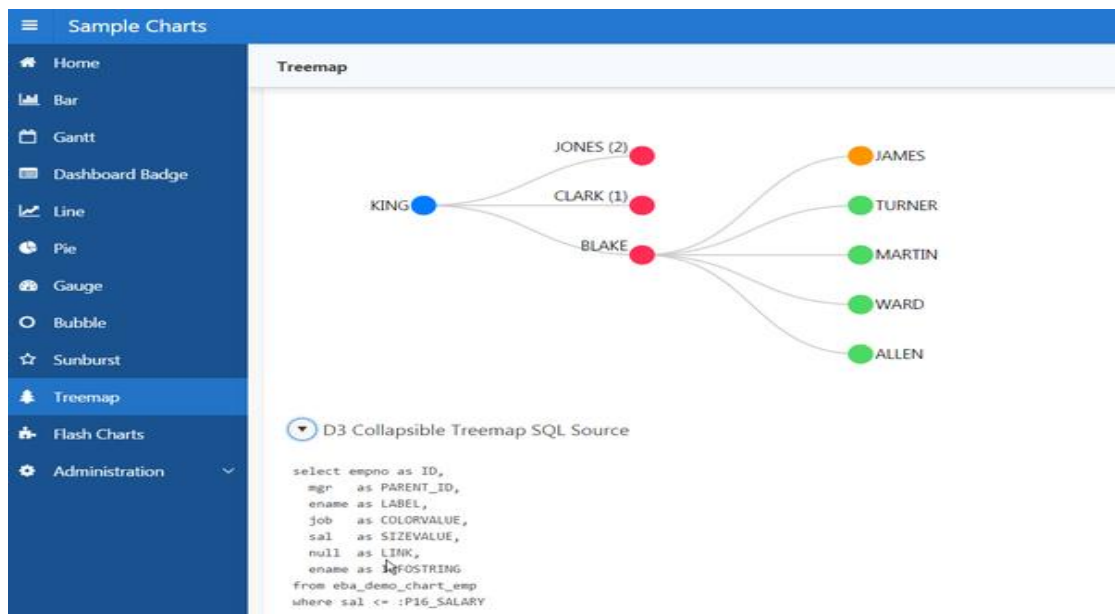


Рисунок 6 - вигляд дерева

В арех є вбудований елемент Tree який дозволяє на основі SQL запиту побудувати справжнє багаторівневе дерево (рисунок 7).

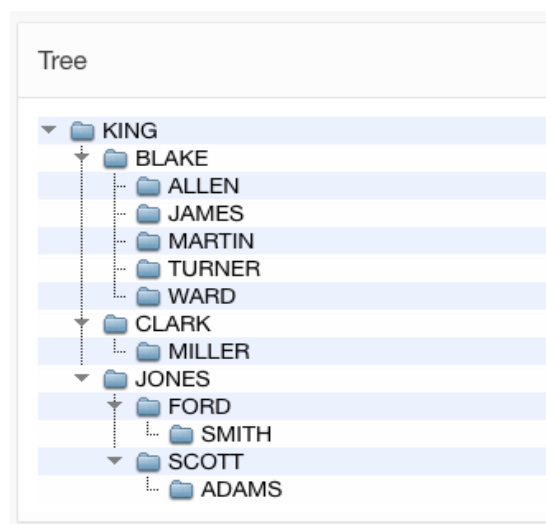


Рисунок 7 — Багаторівневе дерево, створене в Арех

Для побудови такого запиту необхідно створити таблицю, якій повинні міститись такі поля:

- id – значення елемента
- name – текст елемента
- parent – id батьківського елемента.

Запит для побудови дерева виглядає таким чином:

```
select case when connect_by_iseaf = 1 then 0 when level = 1 then 1 else -1 end as status,
level,
name as title,
'icon-tree-folder' as icon,
id as value,
ename as tooltip,
```

null as **link**  
    from **emp**  
    start with **parent** is null  
    connect by prior **id** = **parent**  
    order siblings by **name**

    де title - назва елемента, icon – зображення, value - значення елемента, tooltip – текст підказки, link – адреса куди перенаправляє елемент, emp – таблиця з елементами для побудови дерева, **parent** – id батьківського елемента (для корінного елемента id = null), **id** = **parent** — вираз за яким запит буде дерево, де id – значення елемента parent – значення батьківського елемента, siblings by **name** – поле за яким необхідно сортувати.

## Лабораторна робота № 7. Покращення користувацького інтерфейсу

Для покращення користувацького інтерфейсу встановимо додаток Universal Theme Sample Application з бібліотеки демонстраційних додатків (рис. 1) та на його прикладі розглянемо можливості візуалізації власних проєктів.

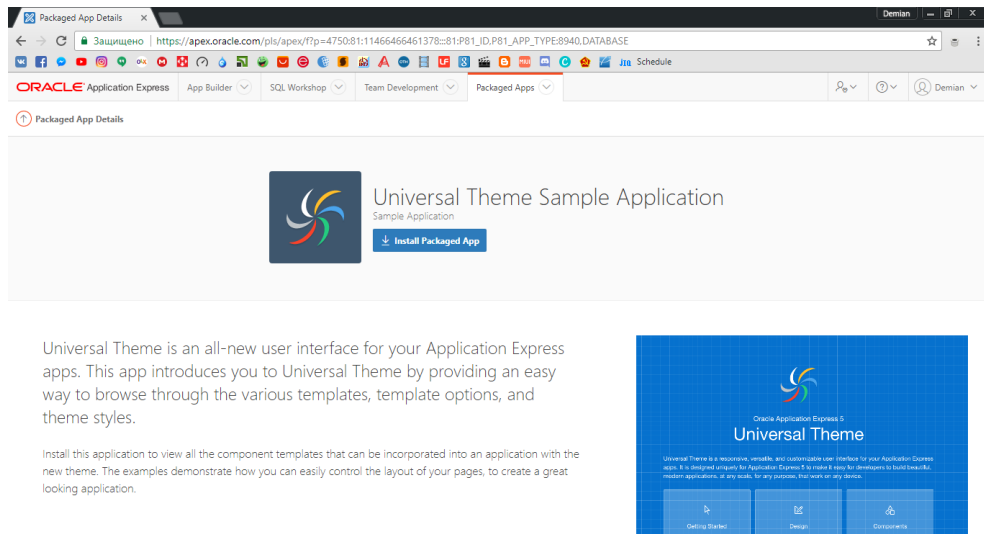


Рисунок 1 – Додаток Universal Theme Sample Application

Вигляд головної сторінки додатку Universal Theme Sample Application зображено на рисунку 2.

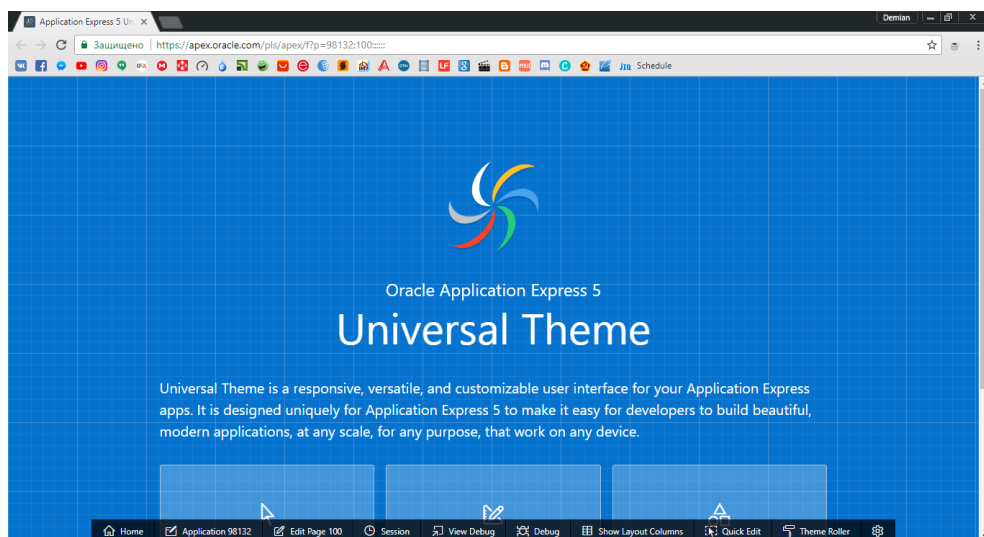


Рисунок 2 – Головна сторінка додатку Universal Theme Sample Application

На вкладці Colors (рис. 3) наведено стандартні кольори, що використовуються для оформлення додатків (15 основних та 30 їх відтінків).

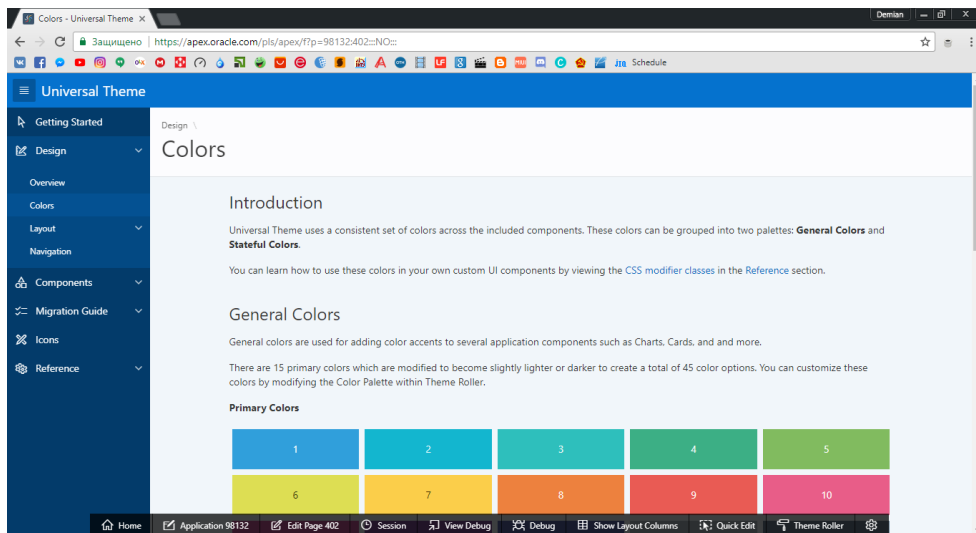


Рисунок 3 – Вкладка Colors

На вкладці Layout (рис. 4) наведена схема можливого розміщення функціональних блоків (розподіл сторінки на регіони).

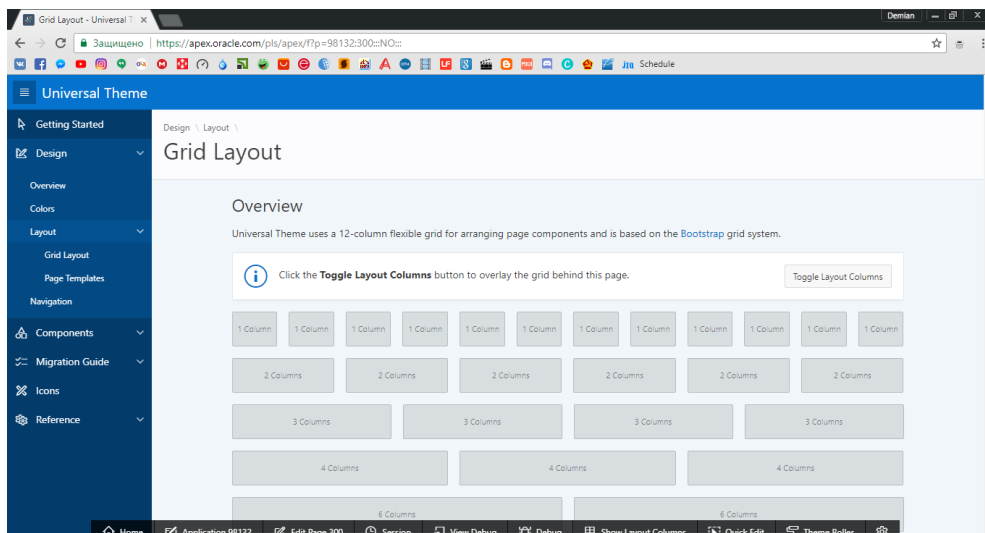


Рисунок 4 – Вкладка Layout

На вкладці Page Templates (рис. 5) наведені шаблони сторінок з різними варіантами розміщення шапок, панелей навігації і тп.

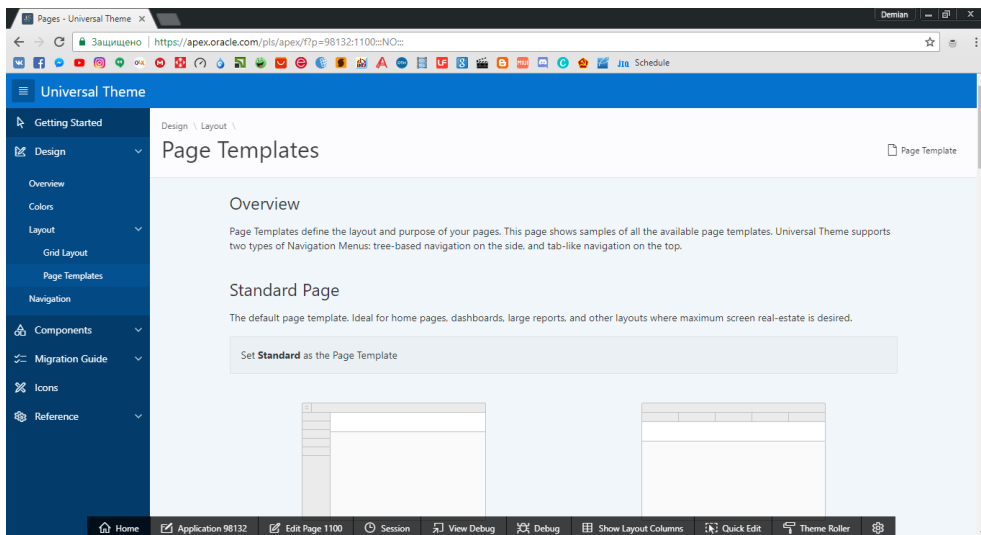


Рисунок 5 – Вкладка Page Templates

На вкладці Timeline (рис. 6) в підменю Components представлено приклади оформлення класичних звітів, що можуть використовуватися у організації подій.

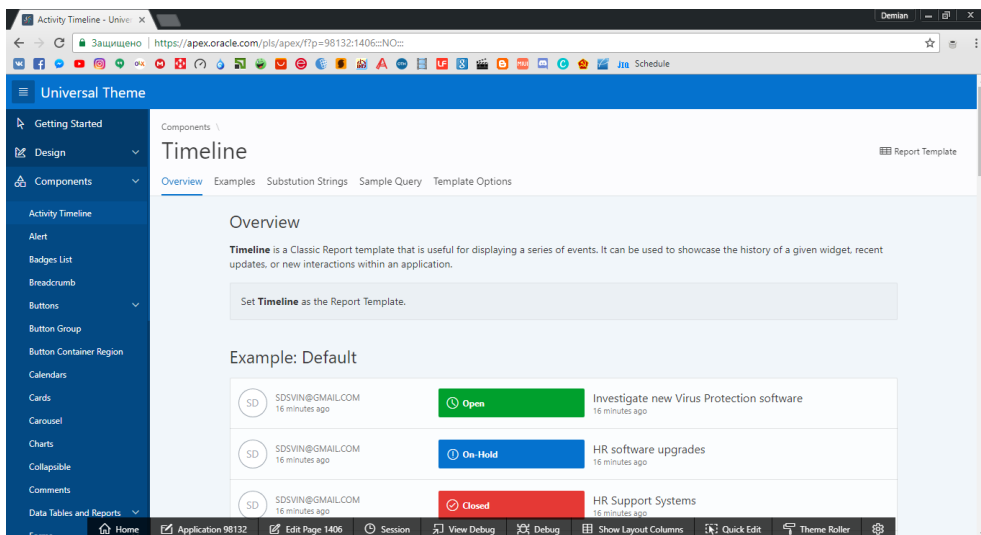


Рисунок 6 – Вкладка Timeline

На вкладці Alert (рис. 7) в підменю Components можна ознайомитися з можливими варіантами оформлення регіонів з різними типами повідомлень (помилка, тривога, підтвердження та ін. дії).



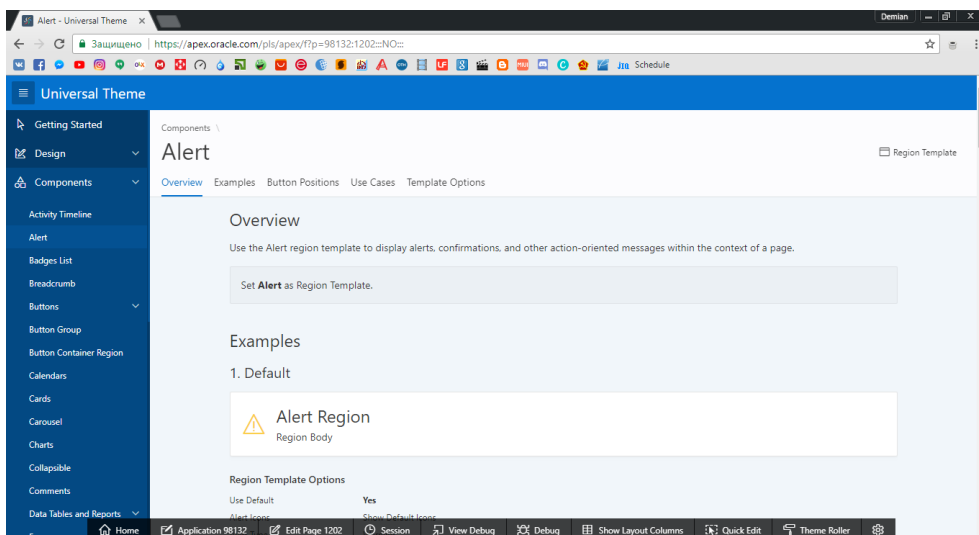


Рисунок 7 – Вкладка Alert

На вкладці Badge List (рис. 8) в підменю Components можна ознайомитися з можливими варіантами оформлення «бейджів», лічильників, тощо.

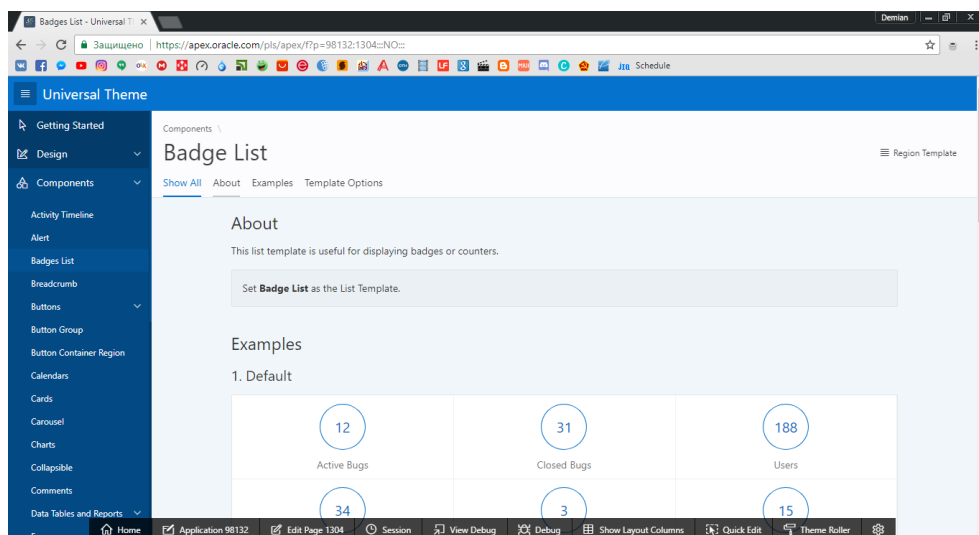


Рисунок 8 – Вкладка Badge List

На вкладці Breadcrumb (рис. 9) в підменю Components наведено можливі варіанти оформлення компонентів навігаційного меню.

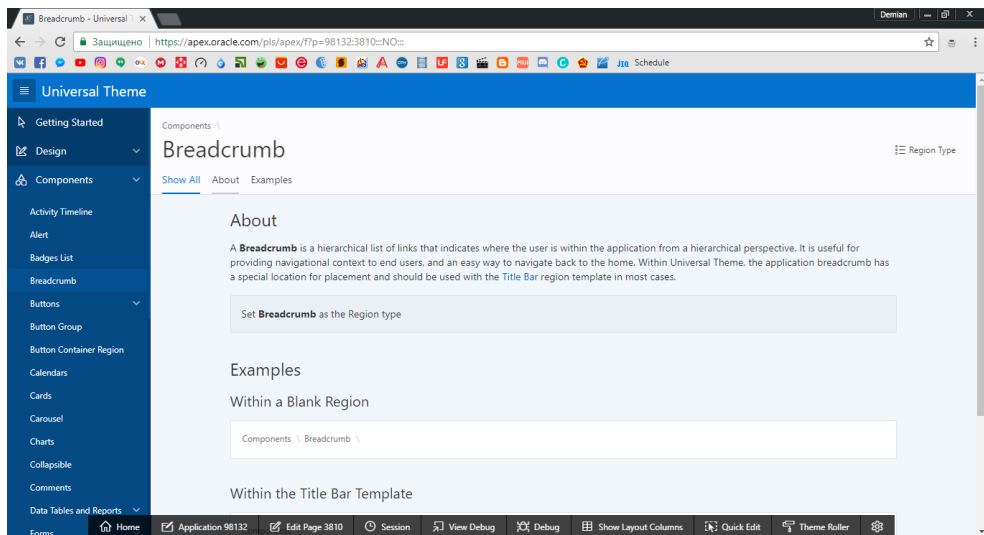


Рисунок 9 – Вкладка Breadcrumb

На вкладці Buttons (рис. 10) в підменю Components можна ознайомитися з різними способами оформлення кнопок (текстових, з піктограмою та компонованих).

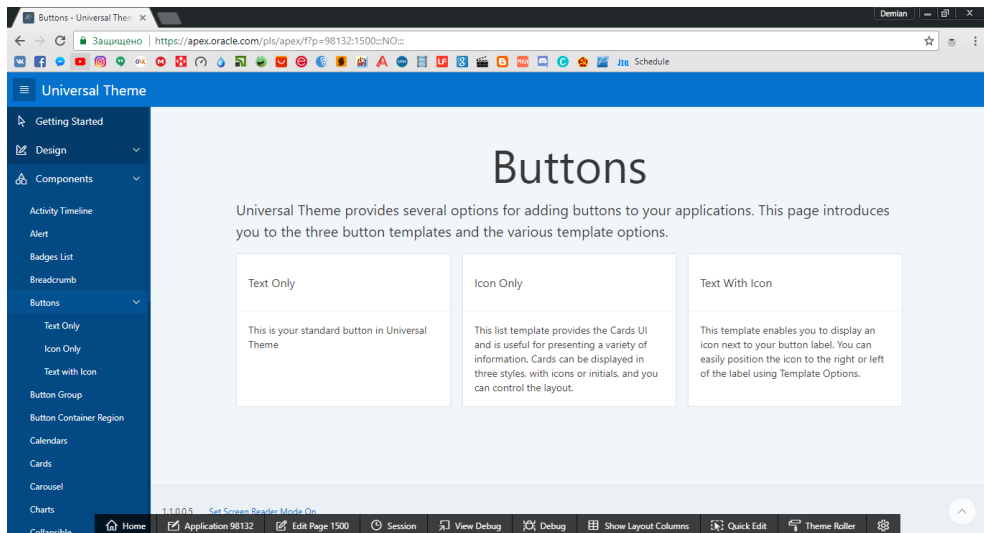


Рисунок 10 – Вкладка Buttons

На вкладці Button Group (рис. 11) в підменю Components для прикладу наведено способи створення груп кнопок.

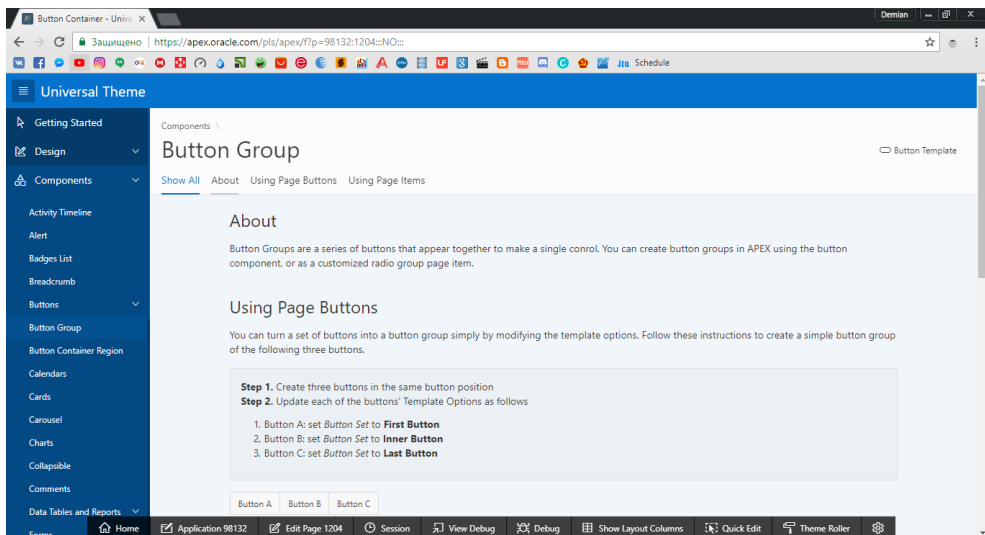


Рисунок 11 – Вкладка Button Group

Піктограми, які можна використовувати для оформлення власних проєктів, знаходять на вкладці Icons (рис. 12).

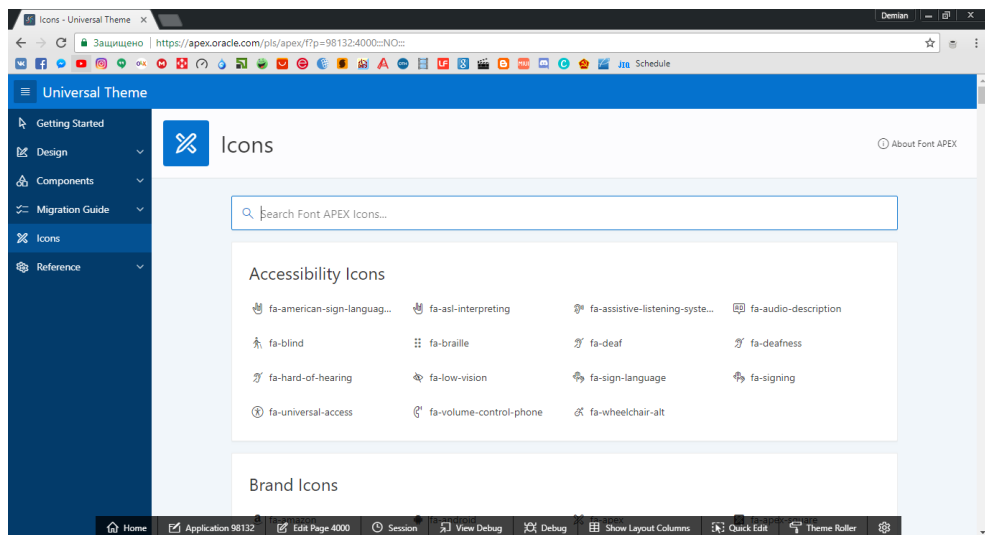


Рисунок 12 – Вкладка Icons

На вкладці Calendar (рис. 13) в підменю Components наведено варіанти оформлення календарів.

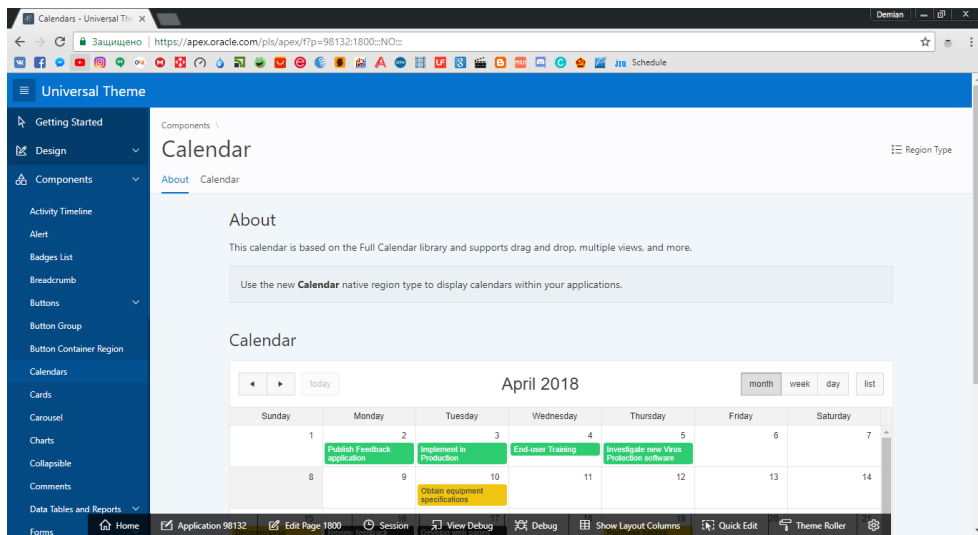


Рисунок 13 – Вкладка Calendar

На вкладці Carousel (рис. 14) в підменю Components наведено варіанти оформлення «каруселей».

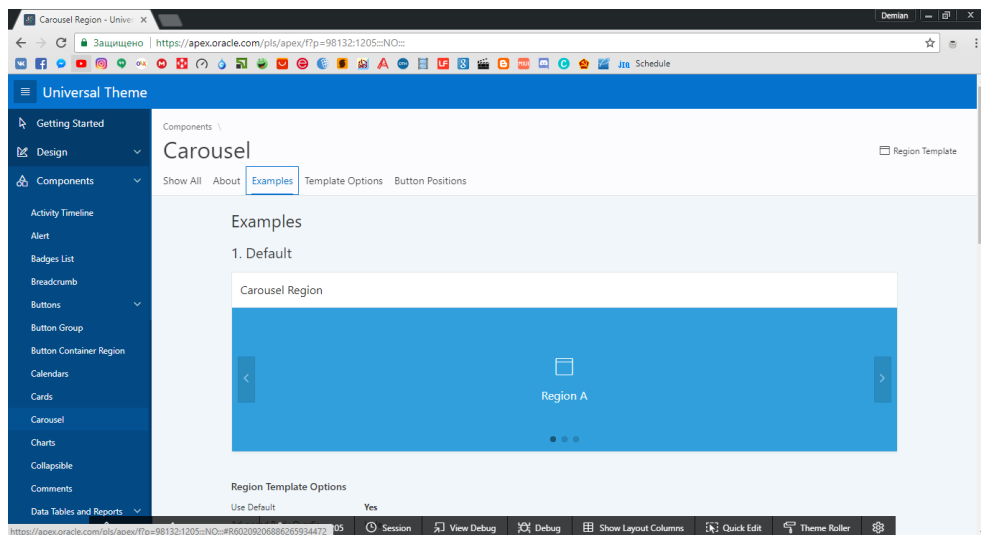


Рисунок 14 – Вкладка Carousel

На вкладці Charts (рис. 15) в підменю Components можна ознайомитися з можливими варіантами оформлення різних типів діаграм та графіків (стовпчастих, кругових, циферблатів і т.п.).

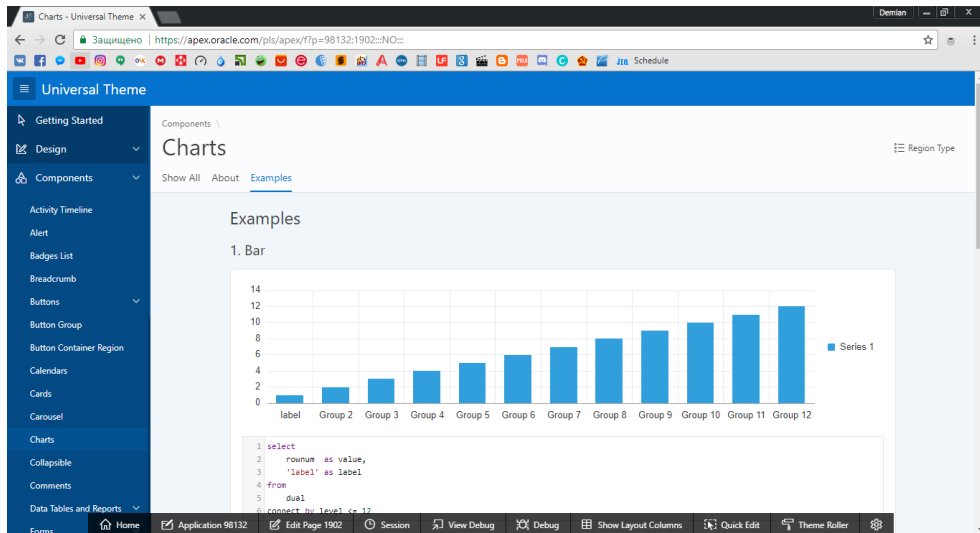


Рисунок 15 – Вкладка Charts

Приклади дерев, які можна використовувати для ієрархічної навігації, знаходять на вкладці Tree (рис. 16) в підменю Components.

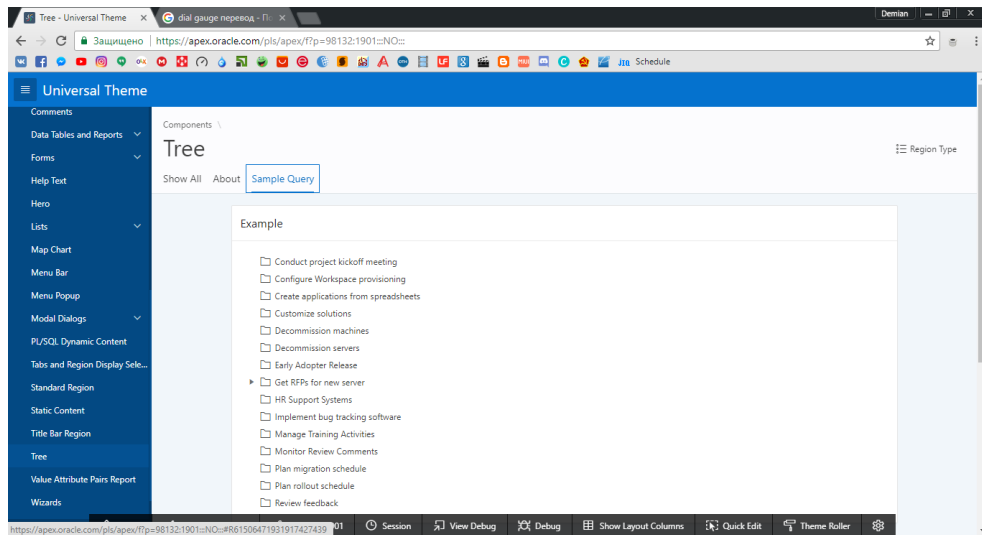


Рисунок 16 – Вкладка Tree

## Лабораторна робота № 8. Експорт імпорт додатків

Для того, щоб створити копію додатку, потрібно зайти у «App Builder», вибрати додаток та натиснути на кнопку редагування рис. 1, після чого обрати «Import / Export» рис. 2. Після чого з'явиться модальне вікно з вибором рис. 3. Обираємо експорт.

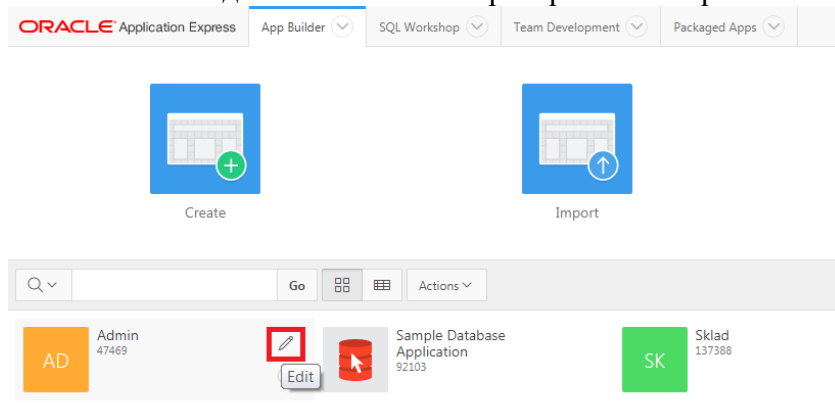


Рисунок 1 – Редагування додатку

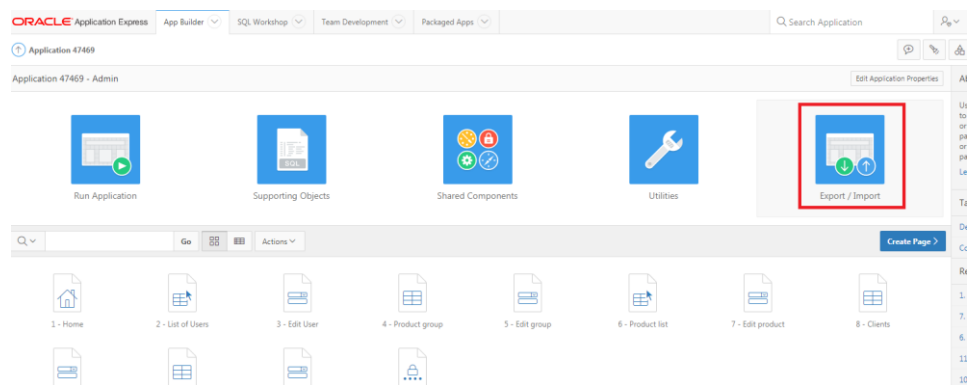


Рисунок 2 – Пункт «Import / Export»

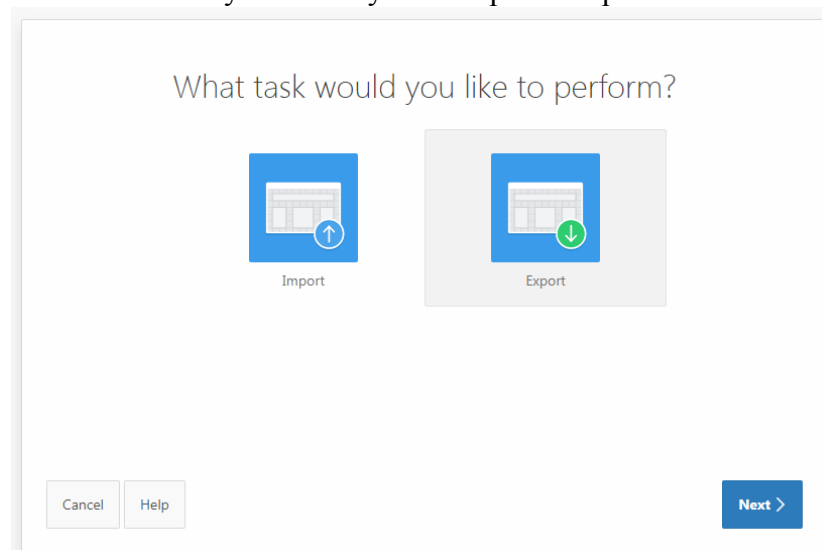


Рисунок 3 – Модальне вікно

Далі потрібно обрати параметри експорту рис. 4. Кожен параметр реалізує:

- File Format:

1. DOS, щоб лінії у результуючому файлі закінчувалися поверненнями каретки та каналами рядка.
2. UNIX, щоб лінії у результуючому файлі містили закінчення каналів рядка.
3. Database, щоб отриманий файл був збережений в Репозиторії Експорту, а не завантажений.

- **Owner Override:**  
За замовчуванням програма належить власнику, ідентифікованому у атрибуті програми. Використовуючи цей атрибут, можна змінити власника.
- **Build Status Override:**
  1. Run Application Only - користувачі можуть запускати лише додаток. Цей параметр призначений для додатків у виробничому примірнику.
  2. Run and Build Application - розробники та користувачі можуть як запускати, так і розробляти додаток

Примітка: Цей параметр можна змінити при імпорті через веб-інтерфейс Application Express.
- **Debugging:**
  1. Так - експортує додаток, увімкнувши налагодження.
  2. Ні - експортує додаток без налагодження.
- **As of:**  
Задає час (у хвилинах) версії коли вона була доступна. Тобто якщо щось пішло не так з додатком, можна отримати додаток, який був у вас кілька хвилин тому.

*Export Preferences потрібні для експорту параметрів користувача.*

*Стандартні налаштування підходять для експорту. Та натискаємо «Export».*

*Після чого почнеться завантаження файлу.*

The screenshot displays the 'Export Application' page in Oracle APEX. At the top, there are navigation tabs: 'Export', 'Workspace', 'Applications' (selected), 'Worksheets', 'Themes', 'Plug-ins', 'User Interface Defaults', and 'Feedback'. The main content area is titled 'Export Application' and includes a 'Reset' button and a highlighted 'Export' button. Below this, there is a 'Choose Application' section with a dropdown menu showing '47469 Admin'. The 'Export Application' section contains several configuration options: 'Selected Application: Admin', 'Page Count: 12', 'Owner: BRUCESHEMA', 'File Format: UNDX', 'Owner Override' (empty), 'Build Status Override: Run and Build Application', 'Debugging: No', 'As of' (empty, with a note 'minutes ago (~ 5 min delay)'), and 'File Character Set: Unicode UTF-8'. The 'Export Preferences' section is expanded, showing a list of options with 'Yes' or 'No' buttons: 'Export Supporting Object Definitions: Yes', 'Export Public Reports: Yes', 'Export Private Reports: Yes', 'Export Report Subscriptions: Yes', 'Export Developer Comments: Yes', 'Export Translations: Yes', and 'Export with Original IDs: Yes'.

Рисунок 4 – Налаштування експорту

Для імпорту, повернемо у «App Builder». Обернемо пункт «Import» рис. 5.

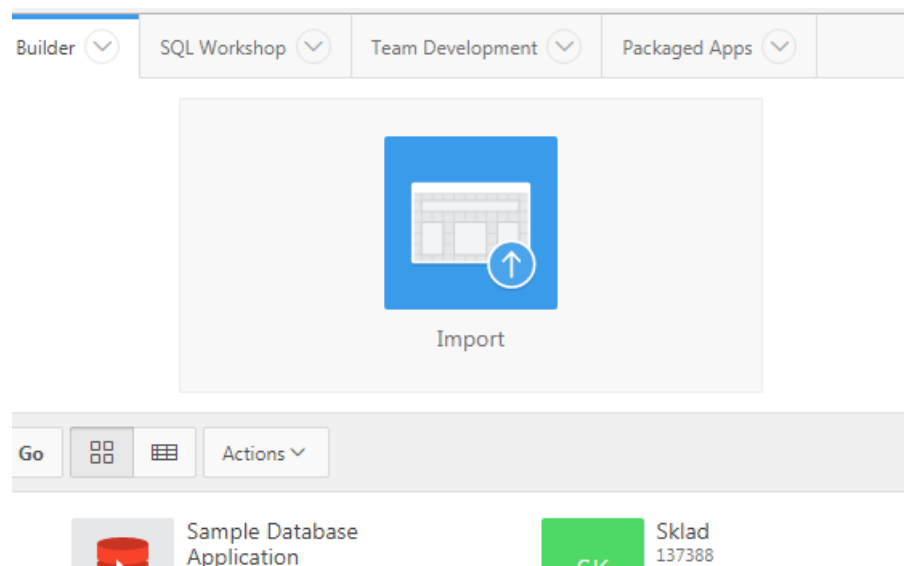


Рисунок 5 – пункт «Import»

Далі обираємо наш попередньо експортований файл рис. 6 та натискаємо «Next».

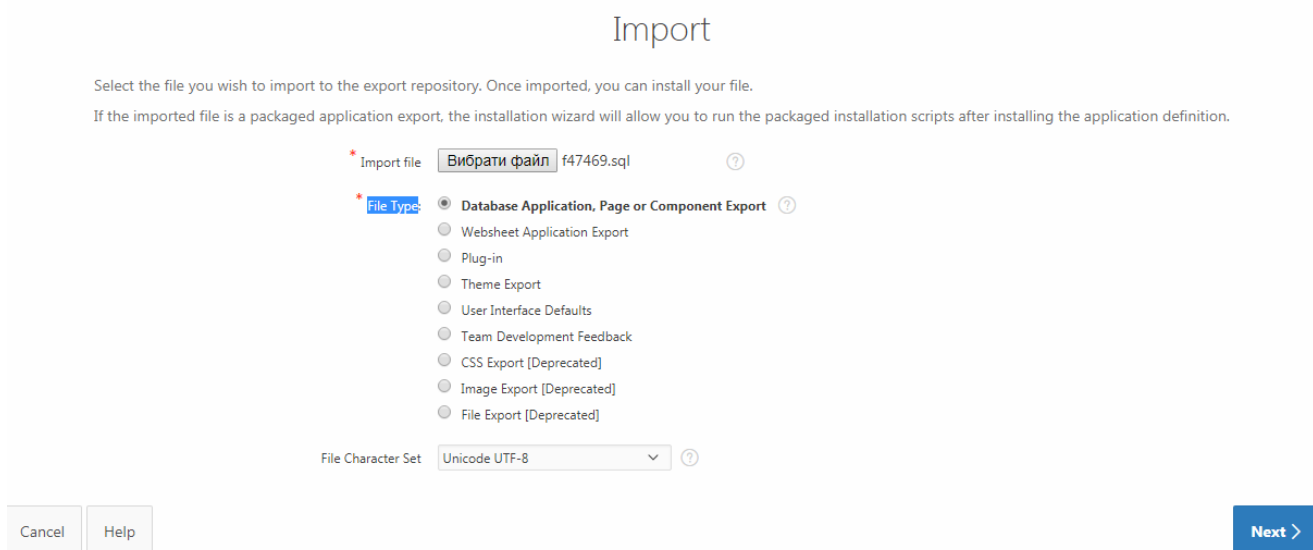


Рисунок 6 – Вибір файлу для імпорту

Другий крок простий і зрозумілий. Третій дозволяє знову задати Build Status Override, та задати ID додаток, тобто згенерувати новий, або для існуючого. Обираємо «Auto Assign New Application ID» для генерування нового ID та натискаємо «Install Application» рис. 7.



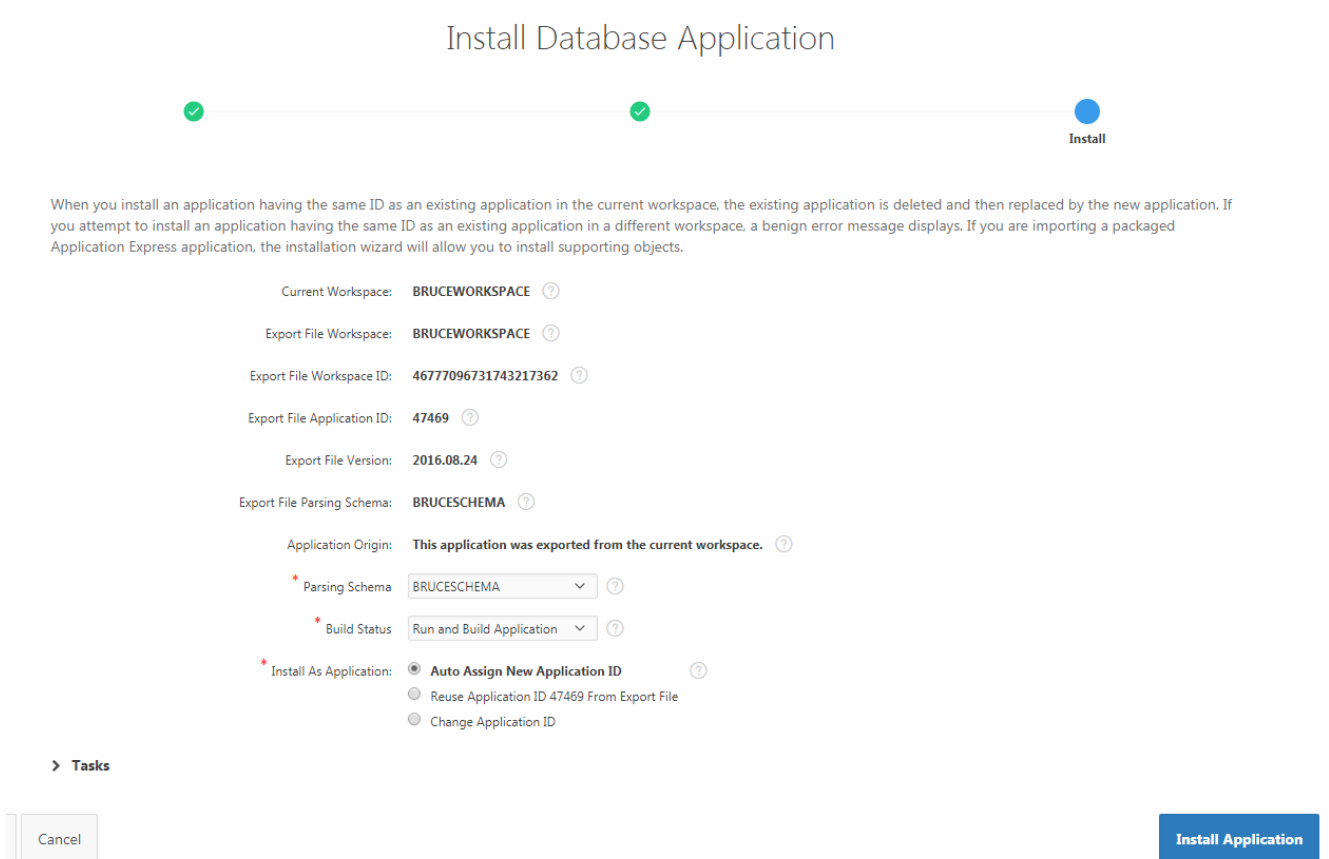


Рисунок 7 – Налаштування імпорту

Потім погоджуємось з додаванням додаткових файлів. Після виконаних дій, буде створено новий додаток рис. 8.

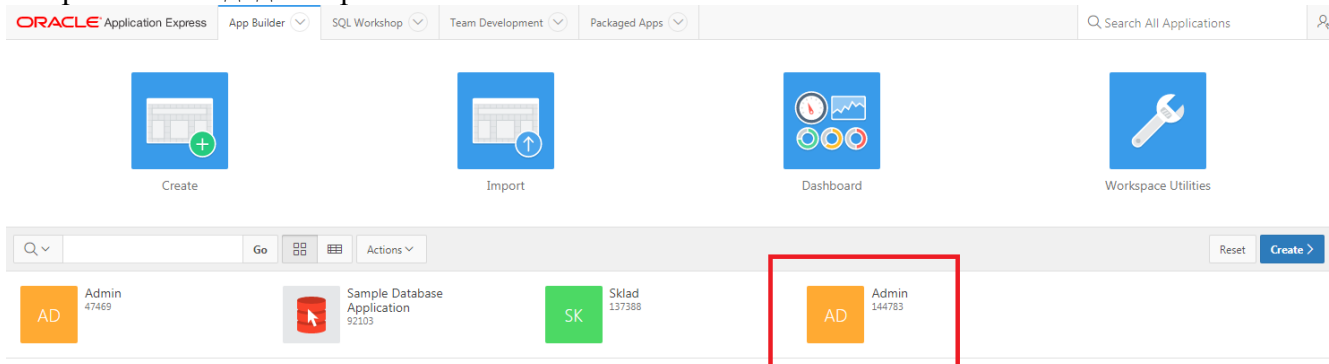


Рисунок 8 – Результат імпорту додатку.

Щоб видалити створений додаток, потрібно увійти до редагування, натиснувши на олівець поряд додатком та видалити його рис. 9.

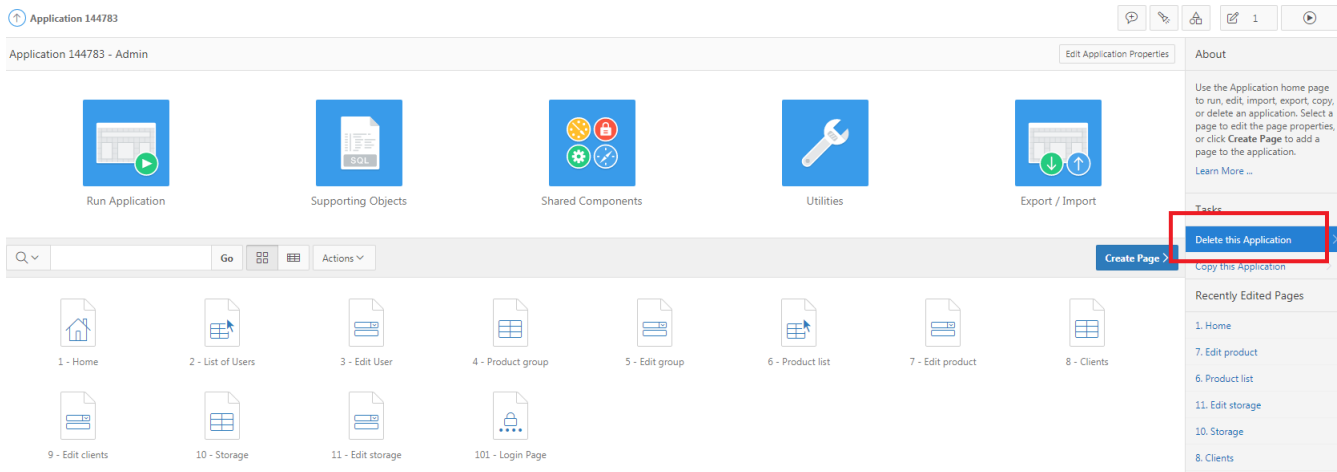


Рисунок 9 – Видалення додатку

## Доопрацювання дерева

У попередніх уроках, ми створили дерево, але воно не працювало, а тільки показувало структуру. Тепер зробимо функціонал.

Додамо приховану змінну, яка буде приймати ID вибраного елемента у дереві та віднімаємо перевірку Arx на цю змінну, щоб не було помилок у роботі змінної рис. 10.

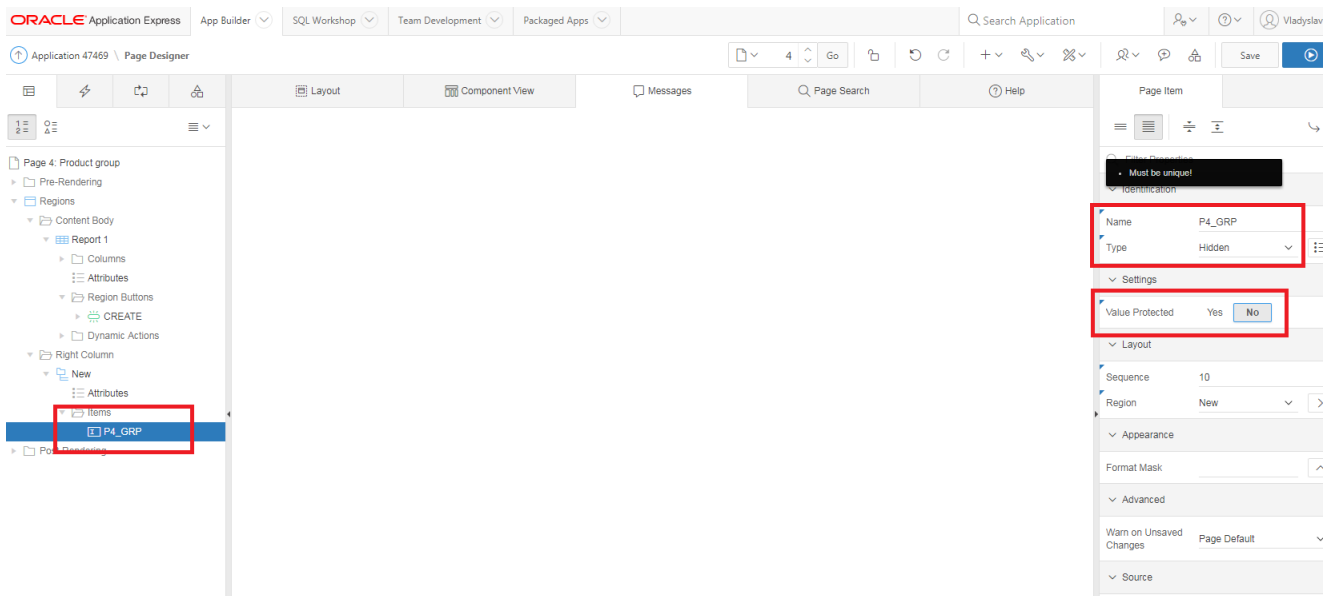


Рисунок 10 – Новий елемент.

Далі нам потрібно редагувати SQL запит у дереві.

Візьмемо url нашої сторінки (на якій розміщене наше дерево), але не повністю рис.

11.

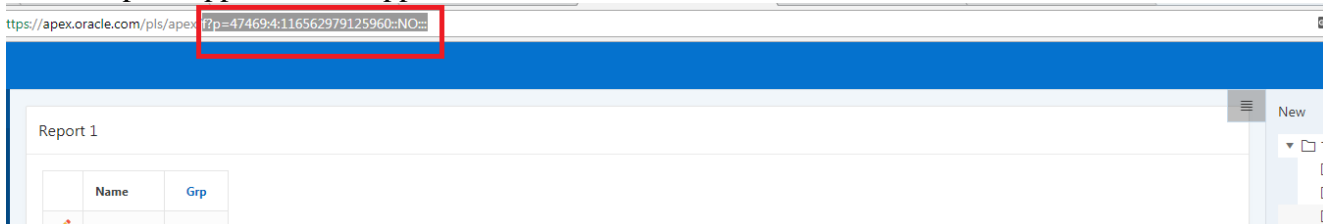
У вас воно буде не таким.

Номер додатку Номер сторіки Номер сесії

f?p=47469:4:116562979125960:.....

для більш універсального запису, змінимо цей URL на

f?p=&APP\_ID.:4: &APP\_SESSION:.....



До цього URL додаємо **назву** нашої змінної, яку ми створили, лапки та наше ID  
'f?p=&APP\_ID.:4:&APP\_SESSION:.....:P4\_GRP:|id

Копіюємо **власну** створену стрічку у SQL запит дерева рис. 11.

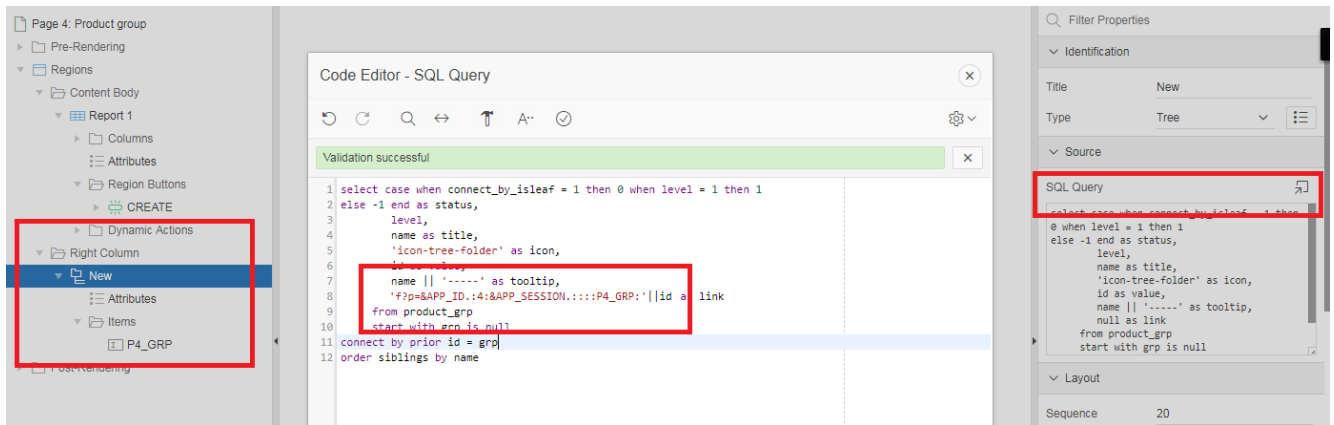


Рисунок 11 – Редагування SQL запиту у деревці

Тепер нам потрібно вказати нашому «Репорту», щоб він брав дані з змінної яка має ID елемента дерева рис. 12.

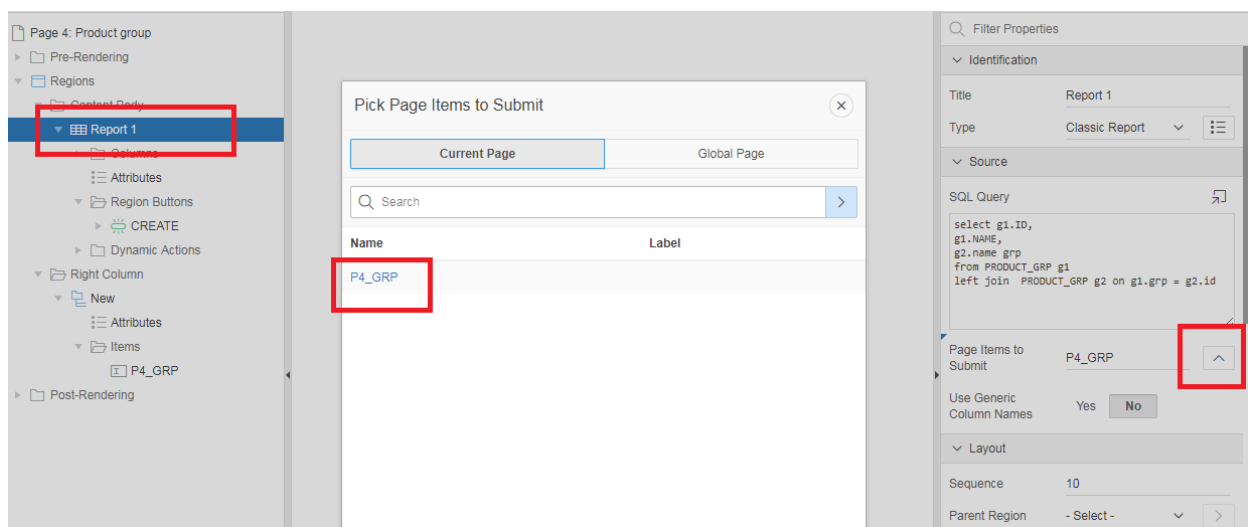


Рисунок 12 – Вибір змінної для «Сабміту»

Редагуємо наш SQL запит Репорта. Додавши:  
 where (:P4\_GRP is null or g1.grp = :P4\_GRP)

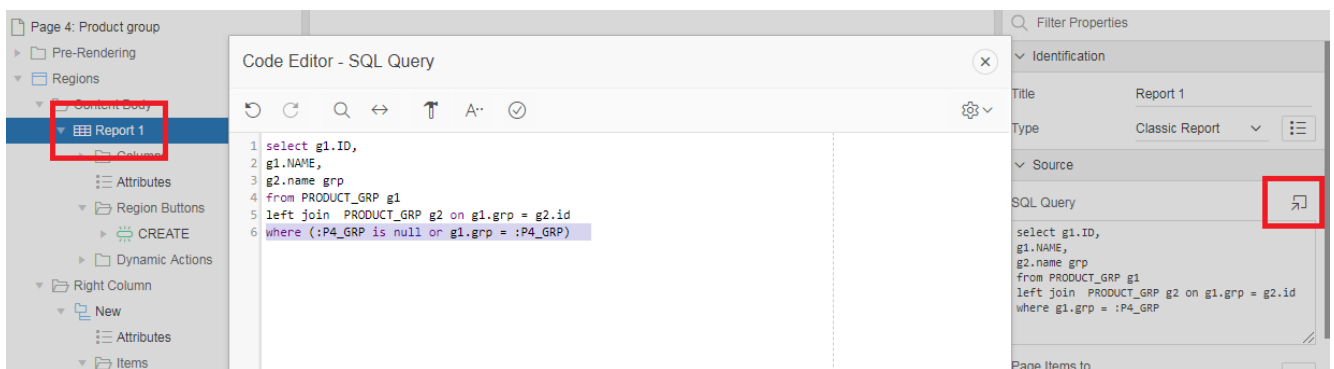


Рисунок 13 – Налаштування репорту

Щоб відмінити постійне згортання дерева після вибору елемента з нього, додамо до атрибуту «Select Node Page Item» нашу приховану змінну. Налаштування дерева зображено на рисунку 14.

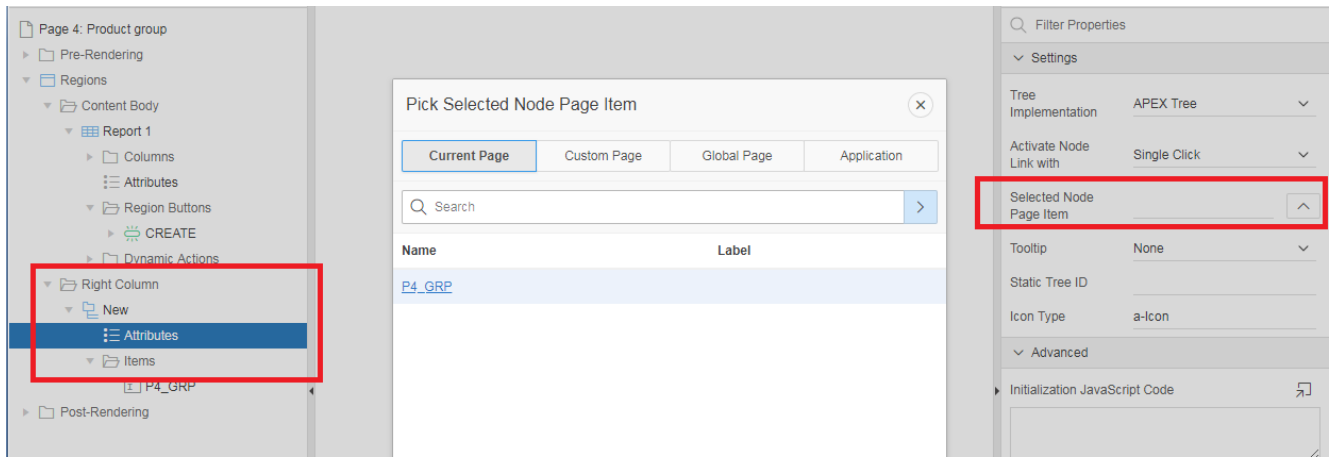


Рисунок 14 – Налаштування дерева

### Розробка додатку «калькулятор»

Нам потрібно завантажити пустий проект рис. 20 з GitHub «Material Apex» з <https://github.com/vincentmorneau/material-apex/releases/tag/v0.12.2>

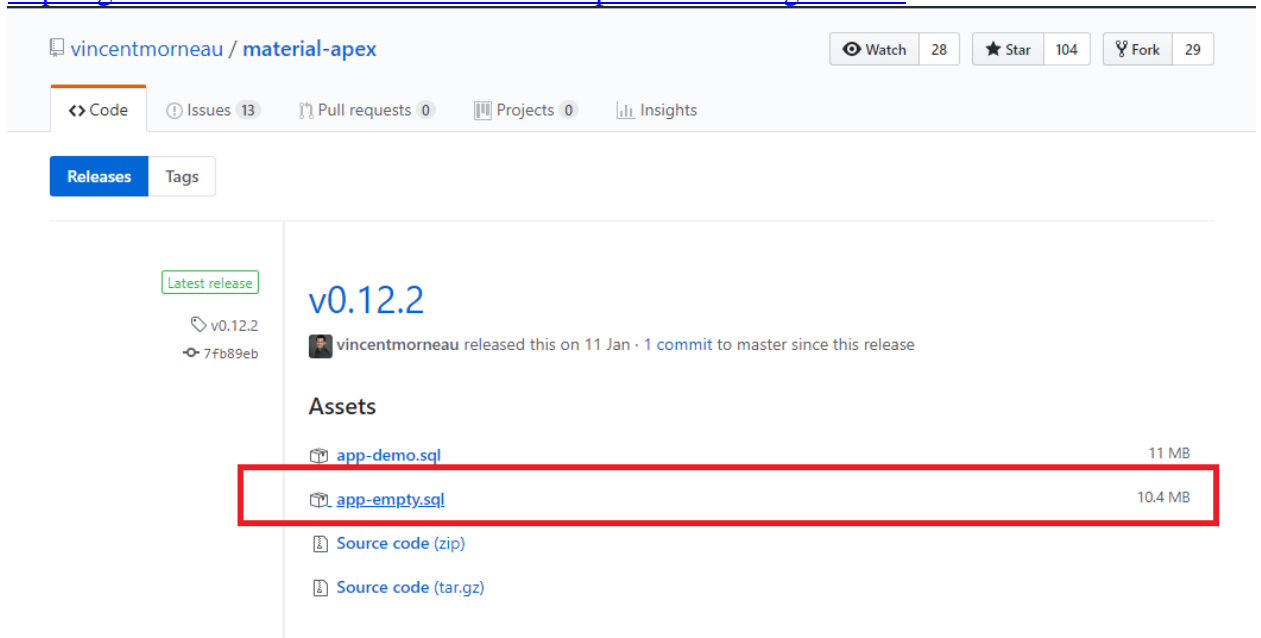


Рисунок 20 – Пустий додаток

Далі імпортуємо щойно завантажений пустий додаток та запускаємо. Вводимо логін та пароль при реєстрації у Apex. Імпортований додаток буде мати вигляд простої сторінки. Вигляд додатку зображено на рисунку 21.

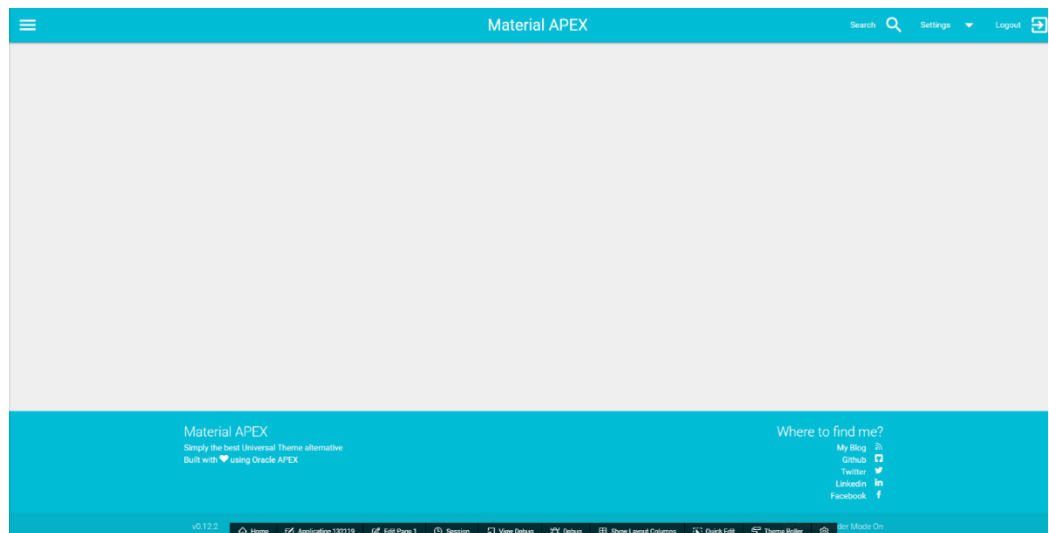


Рисунок 21 – Вигляд додатку

Натискаємо «Edit Page 1» рис. 22, для редагування сторінки.

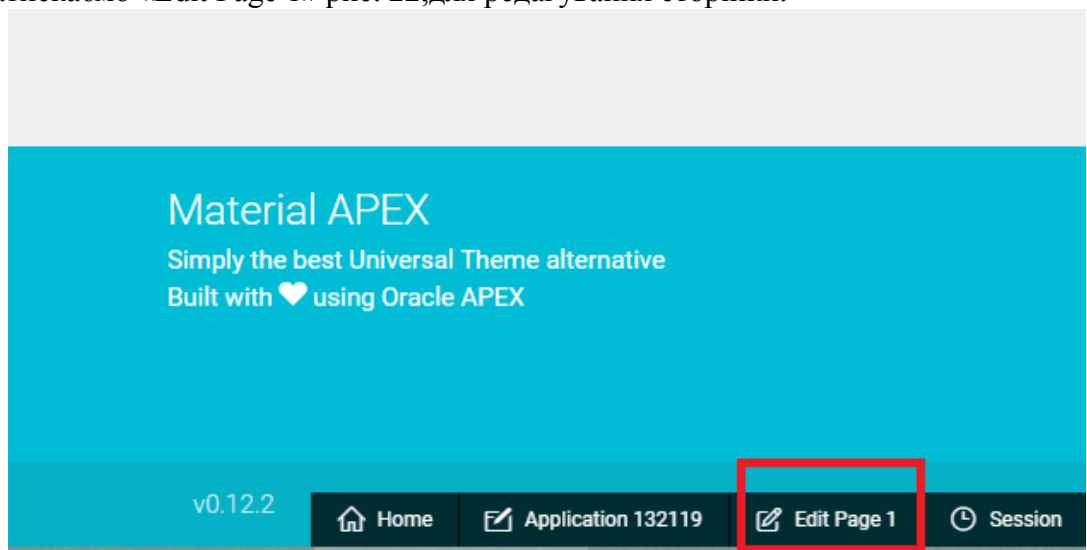


Рисунок 22 – кнопка для редагування сторінки

Додаємо новий регіон з атрибутом Title: Табло, та додаємо «Page Item» рис 23.

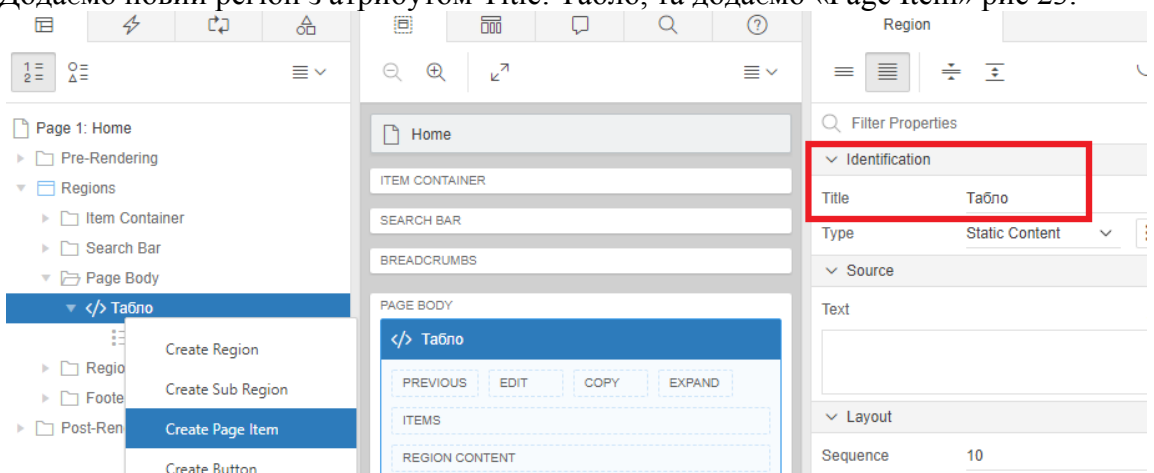


Рисунок 23 – Додавання «Page Item» у новий регіон

Додаємо текстове поле для виводу результатів обрахунків калькулятора. У «File properties» у розділі «Appearance» натискаємо на «Use template defaults» та ставимо галочку на «Hide Label» рис. 24.

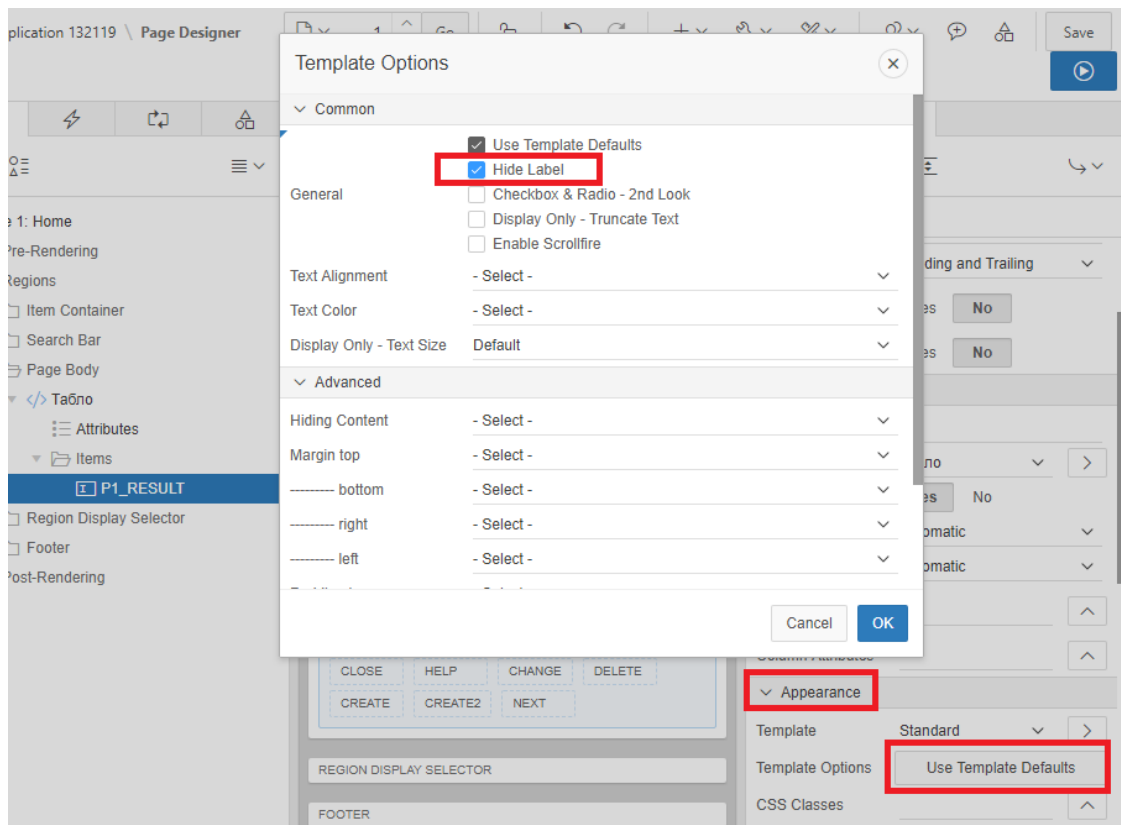


Рисунок 24 – Зміна параметрів текстового поля

Далі додаємо новий регіон «Кнопки» та ховаємо його заголовок рис. 25.

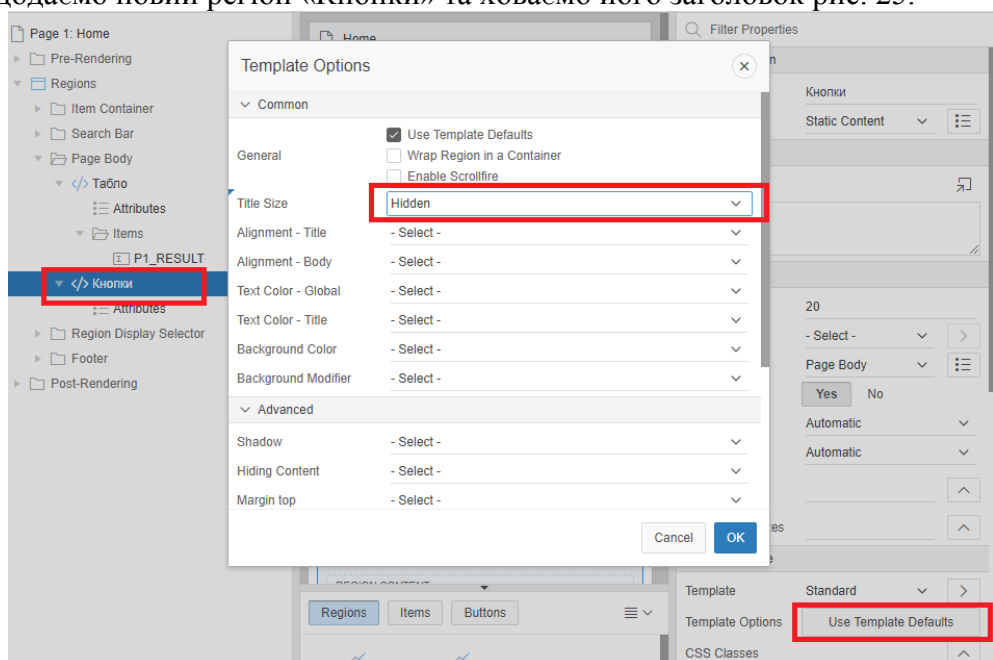


Рисунок 25 – Додавання та налаштування нового регіону

Додаємо перший ряд кнопок. Кожній кнопці задаємо атрибут Button Position: Region Body та встановлюємо значення атрибута Start New Row: No. рис. 26. Це дозволить розташувати кнопки у 1 ряд. Щоб зробити кнопку на всю ширину колонки, потрібно у кожній кнопці поставити галочку на атрибут «Full Column Width» рис. 27.

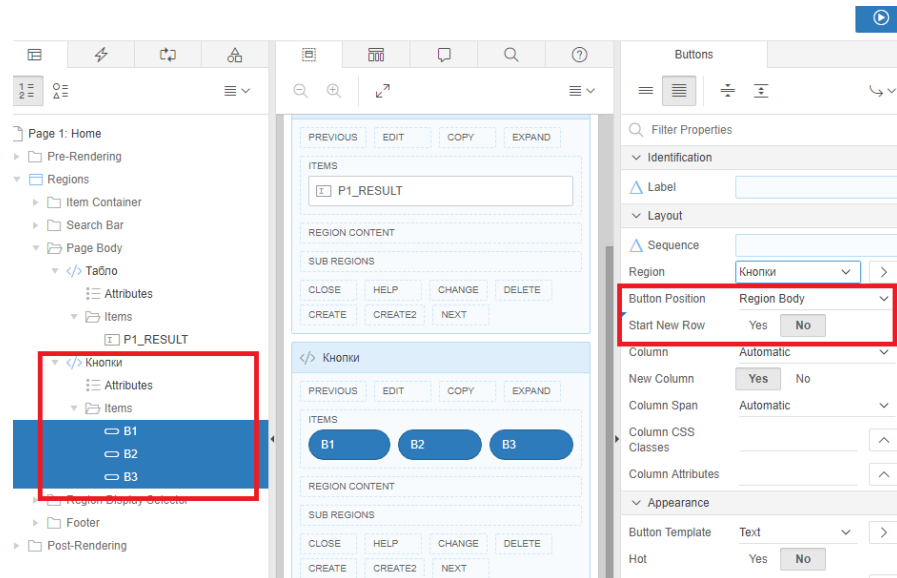


Рисунок 26 – Налаштування кнопок

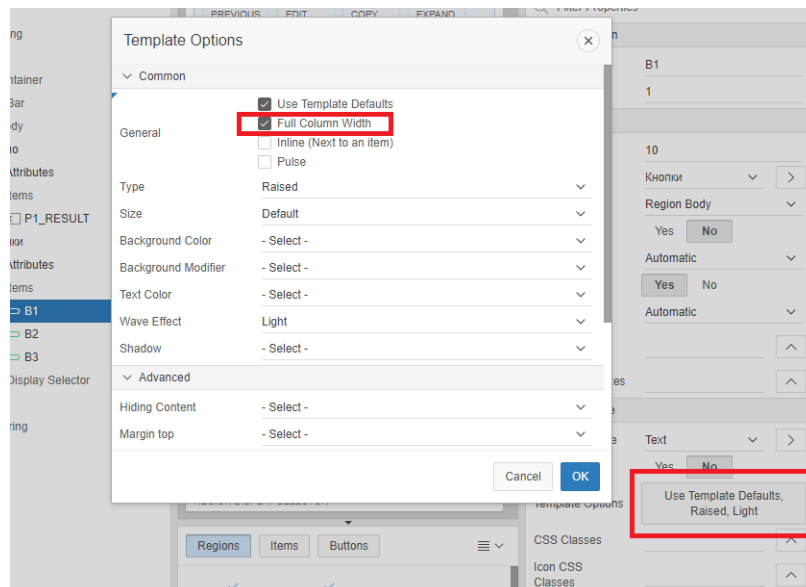


Рисунок 27 – Налаштування кнопки

Прописуємо логіку для калькулятора. Додаємо «Dynamic Actions» до кнопки та назвемо його на приклад «A1». Змінюємо атрибути на:

**Identification**

**Action: Set Value**

**Settings**

**Set Type: PL/SQL Expression**

**Записуємо команду :P1\_RESULT||'1'**

**Items to Submit: P1\_RESULT**

**Affected Elements**

**Item(s): P1\_RESULT**

Для інших кнопок робимо теж саме, тільки змінивши команду PL/SQL Expression на :P1\_RESULT||'%', де замість % - цифра, яка має бути зображена після натискання відповідної кнопки калькулятора та додаємо кнопки дій над числами де теж замість % пишемо \* / + - рис. 28.



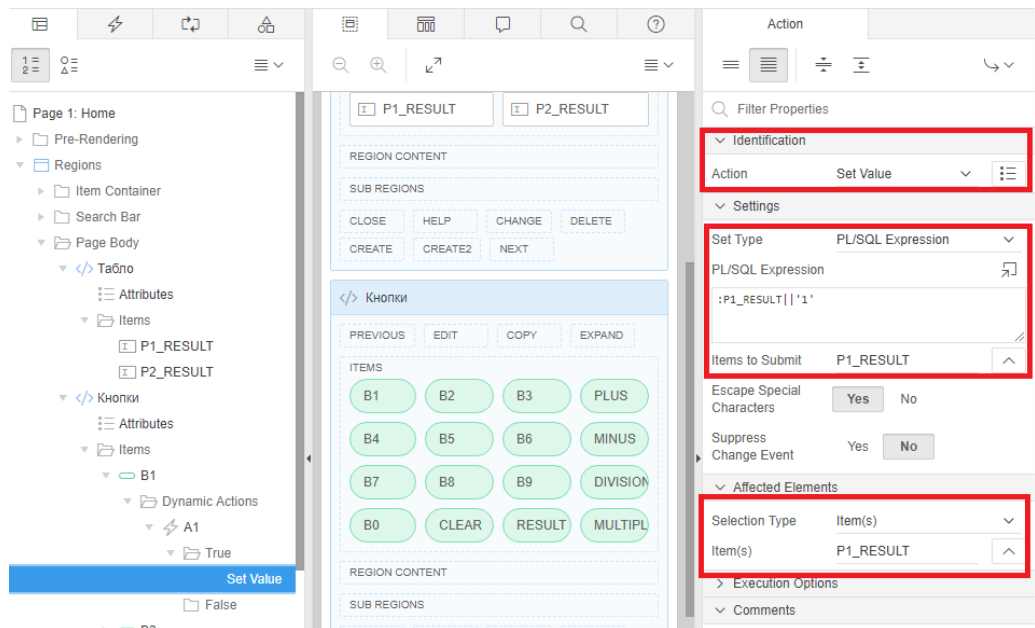


Рисунок 28 – Налаштування кнопки

Для виведення результату та видалення попередньо введеного прикладу додаймо кнопки «=>» та «Clear».

Додаємо ще одне поле «Text Field» для виведення відповіді, назвемо його «P2\_RESULT». Для налаштування кнопки «=>», потрібно змінити атрибути:

**Identification**

**Action: Set Value**

**Settings**

**Set Type: PL/SQL Expression**

**Записуємо команду**

```

declare
v_res number(19,6);
begin
execute immediate 'begin :1 := ' || :P1_RESULT ||'; end;' using out v_res;
returnv_res;
exception when others then
return null;
end;

```

**Items to Submit: P1\_RESULT**

**Affected Elements**

**Item(s): P2\_RESULT**

**Змінені атрибути кнопки наведено на рис. 29.**

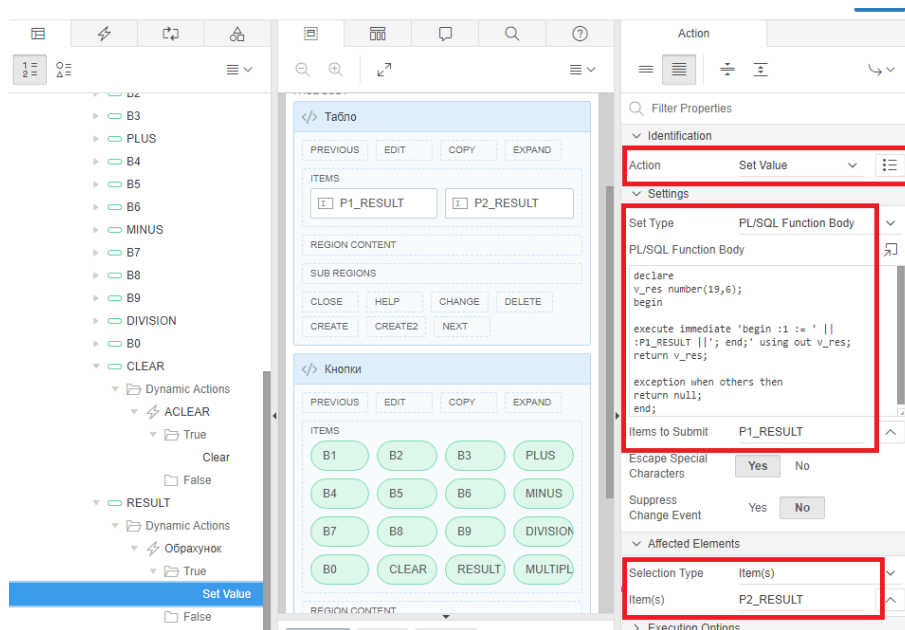


Рисунок 29 – Змінені атрибути

Для налаштування кнопки «Clear» потрібно змінити наступні атрибути:

**Identification**

**Action:** Clear

**Affected Elements**

**Item(s):** P1\_RESULT

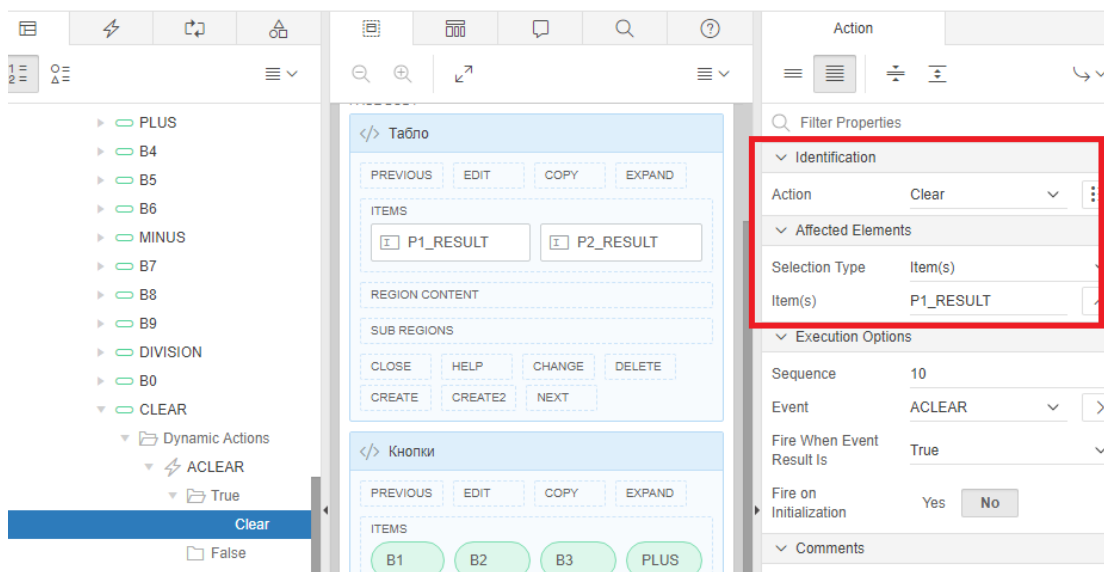


Рисунок 30 – Змінені атрибути

Виконавши все, калькулятор буде мати вигляд як на рисунку 31.



Рисунок 31 – Вигляд калькулятора

## Лабораторна робота № 10. Створення бота

Для створення бота, потрібно зареєструватись на сайті <https://new.corezoid.com/>. Зареєструвавшись, заходимо у профіль рис. 1 та створюємо нову папку рис. 2.

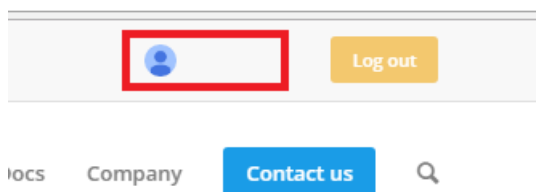


Рисунок 1 – Кнопка входу у профіль

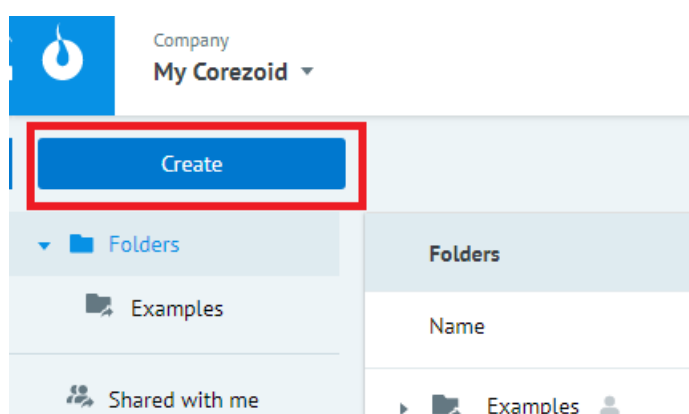


Рисунок 2 – Кнопка створення папки

Назвемо папку «MyBot». Далі створюємо процес «Main» рис. 2.

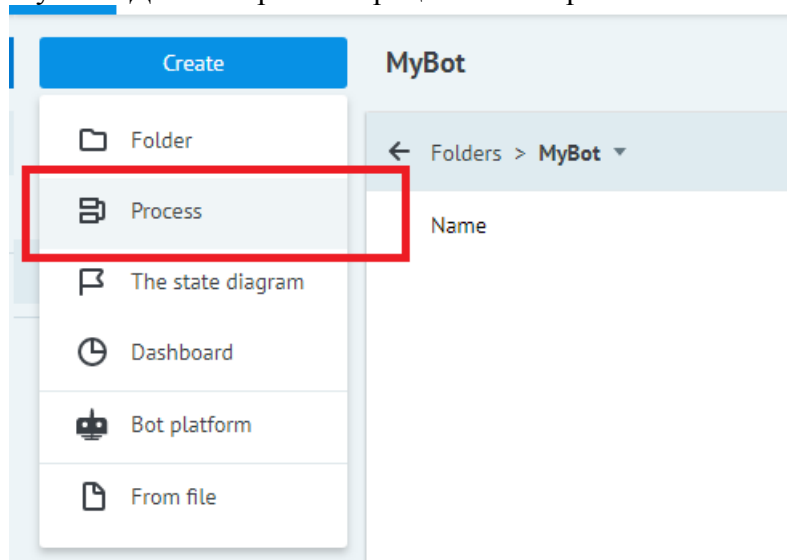


Рисунок 2 – Створення процесу

Заходимо у телеграм. Шукаємо бота на ім'я «BotFather». Створюємо бота командами, які пропонує бот рис. 3, виділене червоним, це наш токен, який потрібне для підключення бота у Corezoid.

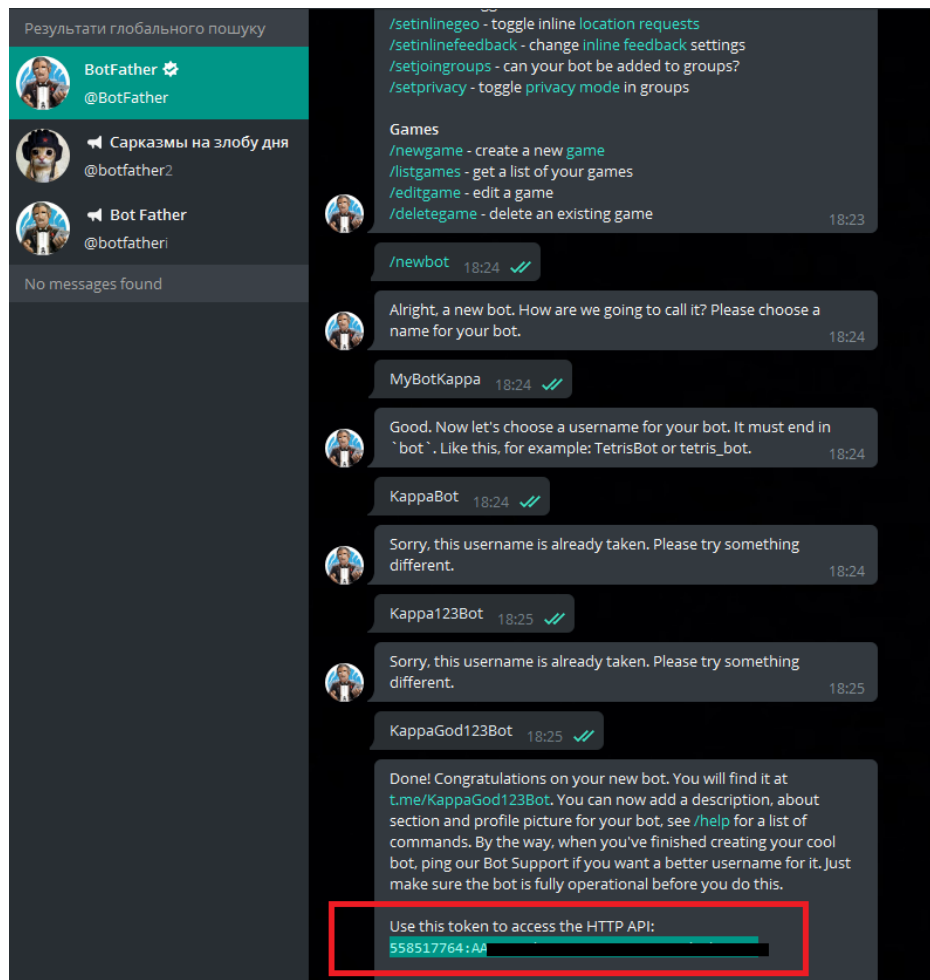
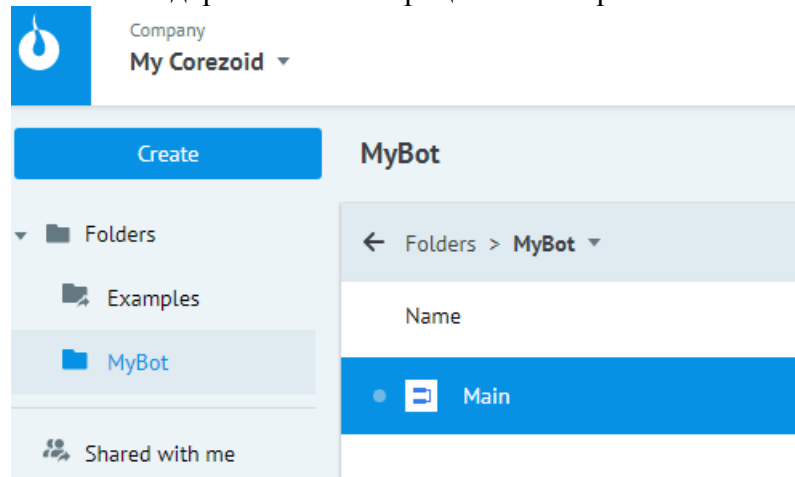


Рисунок 3 – Створення бота у телеграмі

Повертаємось у Corezoid. Відкриваємо наш процес «Main» рис 4.



Далі натискаємо двічі на кнопку старт, та вводим наш токен, який ми отримали при створенні рис. 4 та рис 5.

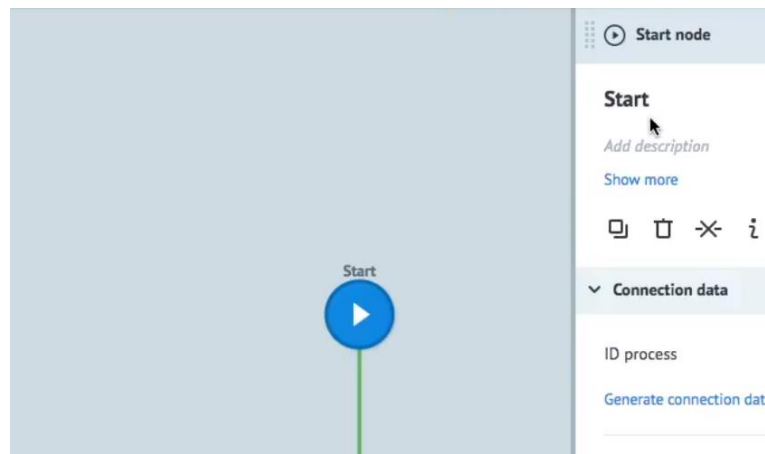


Рисунок 4 – Кнопка старт

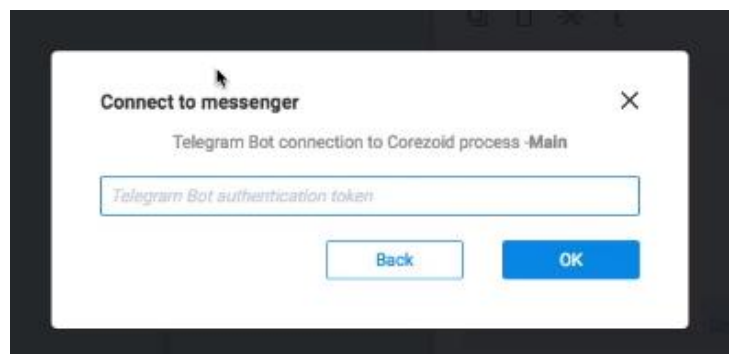


Рисунок 5 – Вікно для введення точена

Якщо все уроблено правильно Corezoid повідомить про це рис 6.

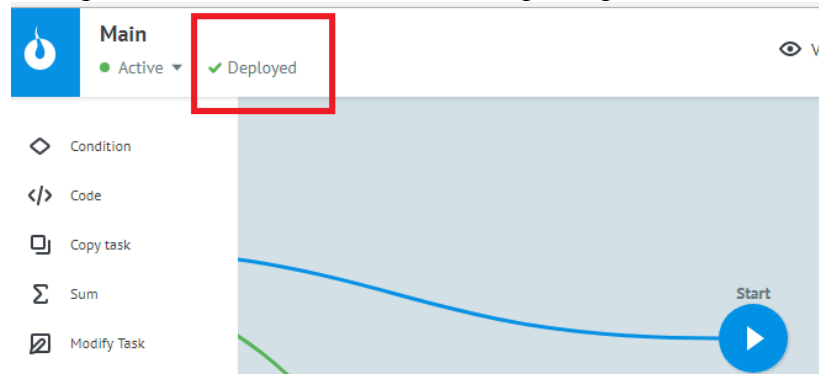
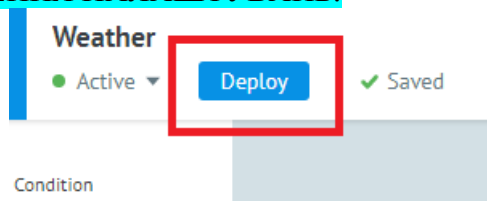


Рисунок 6 – Стан підключення до бота

**ВАЖЛИВО ПІСЛЯКОЖНОЇ ЗМІНИ ПАРАМЕТРІВ НАТИСКАЙТЕ КНОПКУ «DEPLOY» ДЛЯ ЗБЕРЕЖЕННЯ НАЛАШТУВАНЬ.**



Якщо ми щось напишемо боту, то це буде видно у розділі «View» рис. 7.

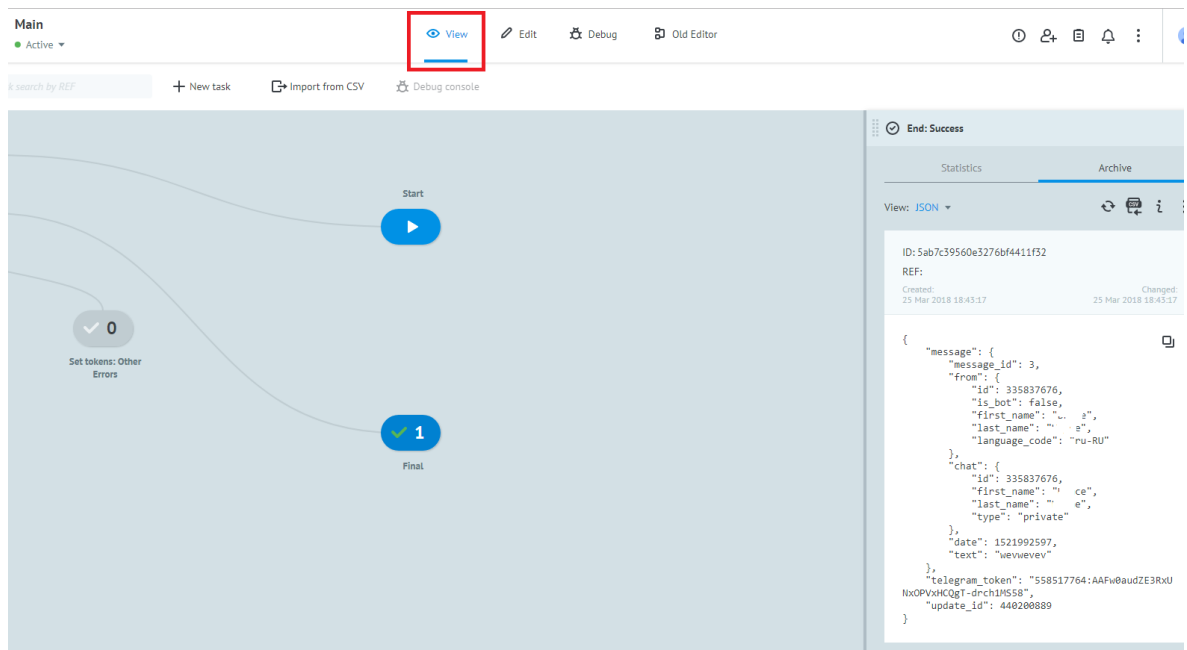


Рисунок 7 – Результат виведення JSON файлу, який передався у процес з телеграма

Додаємо умову на нашу гілку «Final» рис.8. Ця умова буде окремо приймати повідомлення які містять спеціальні слова рис. 8 та рис. 9.

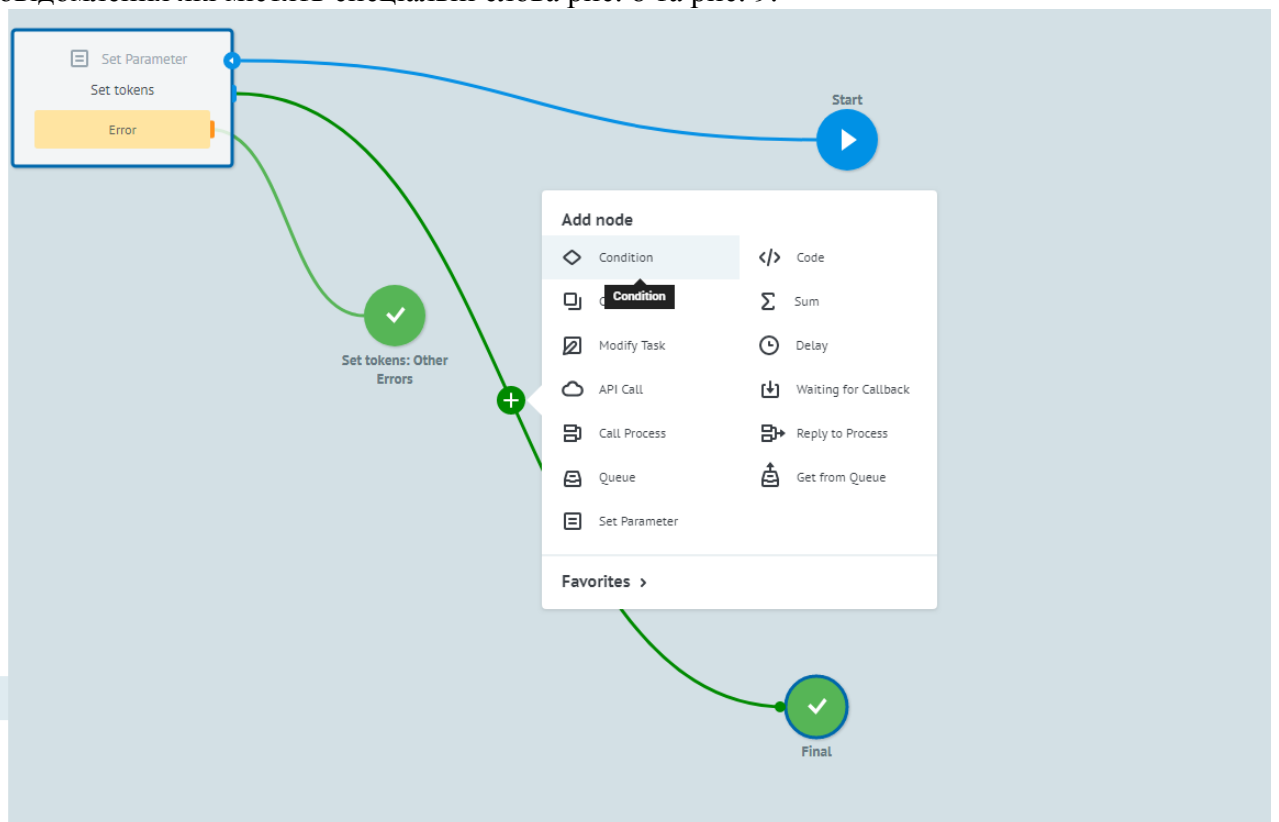


Рисунок 8 – додавання умов

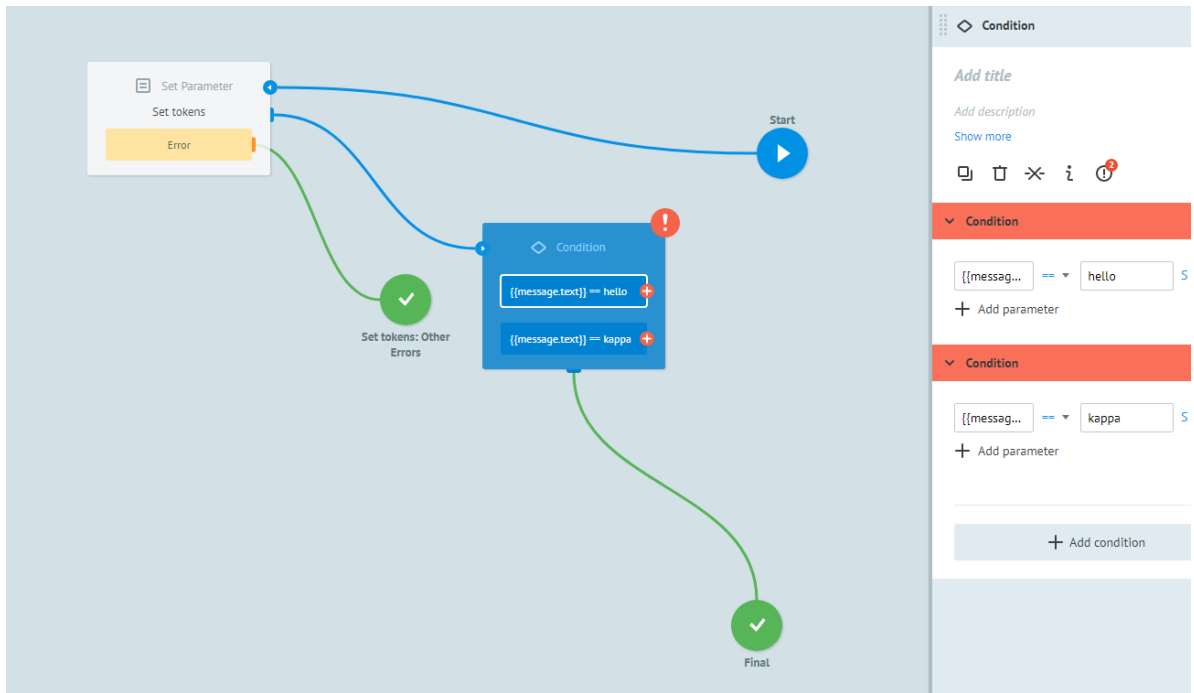


Рисунок 9 – Встановлення умов

Додаємо до наших умов вихід після вдалого виконання умови рис. 10.

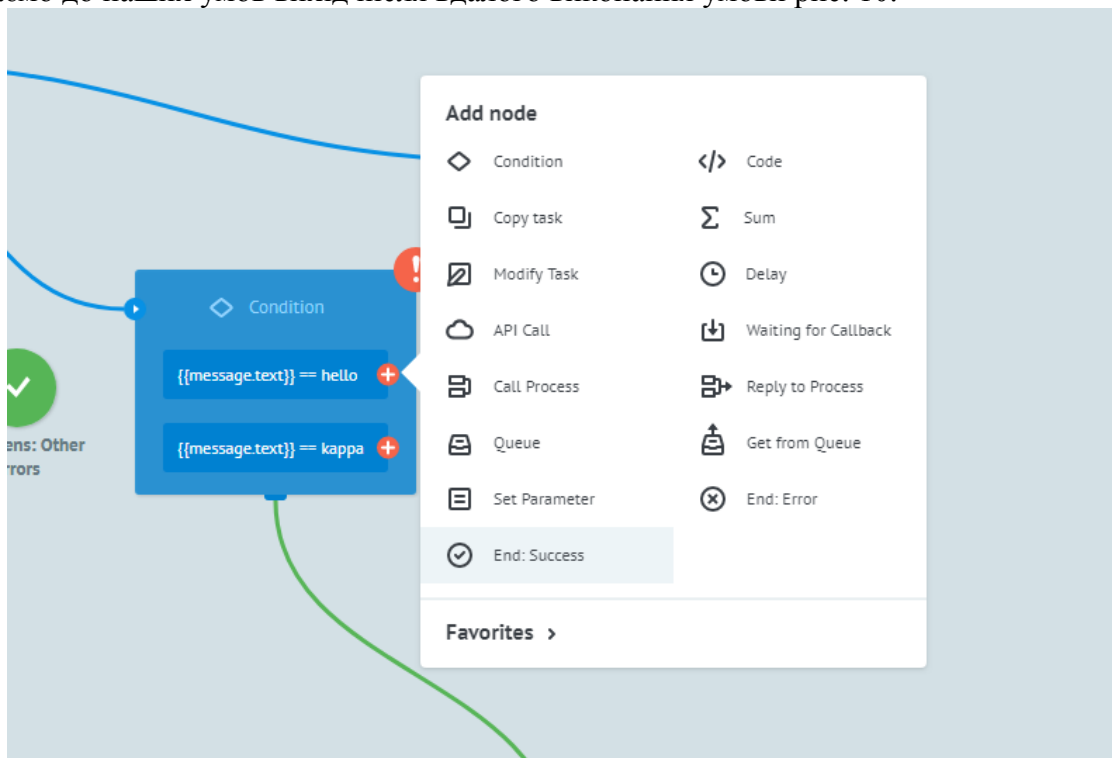


Рисунок 10 – Додавання виходу після виконання умов

Вводимо наші слова ботові, які ми задавали у «умові». Результат наведено на рисунках 11 та 12.

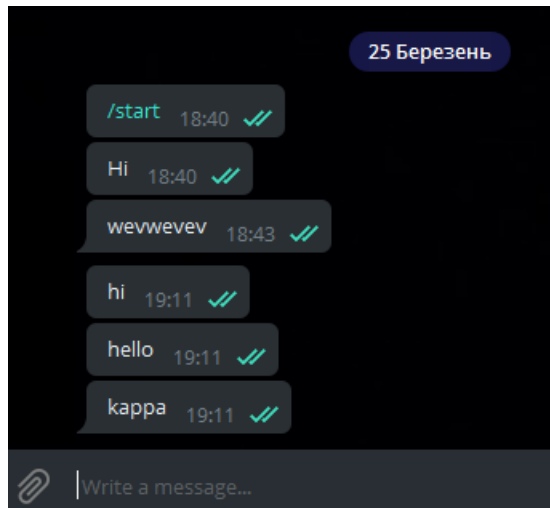


Рисунок 11 – Введені зарезервовані слова ботів

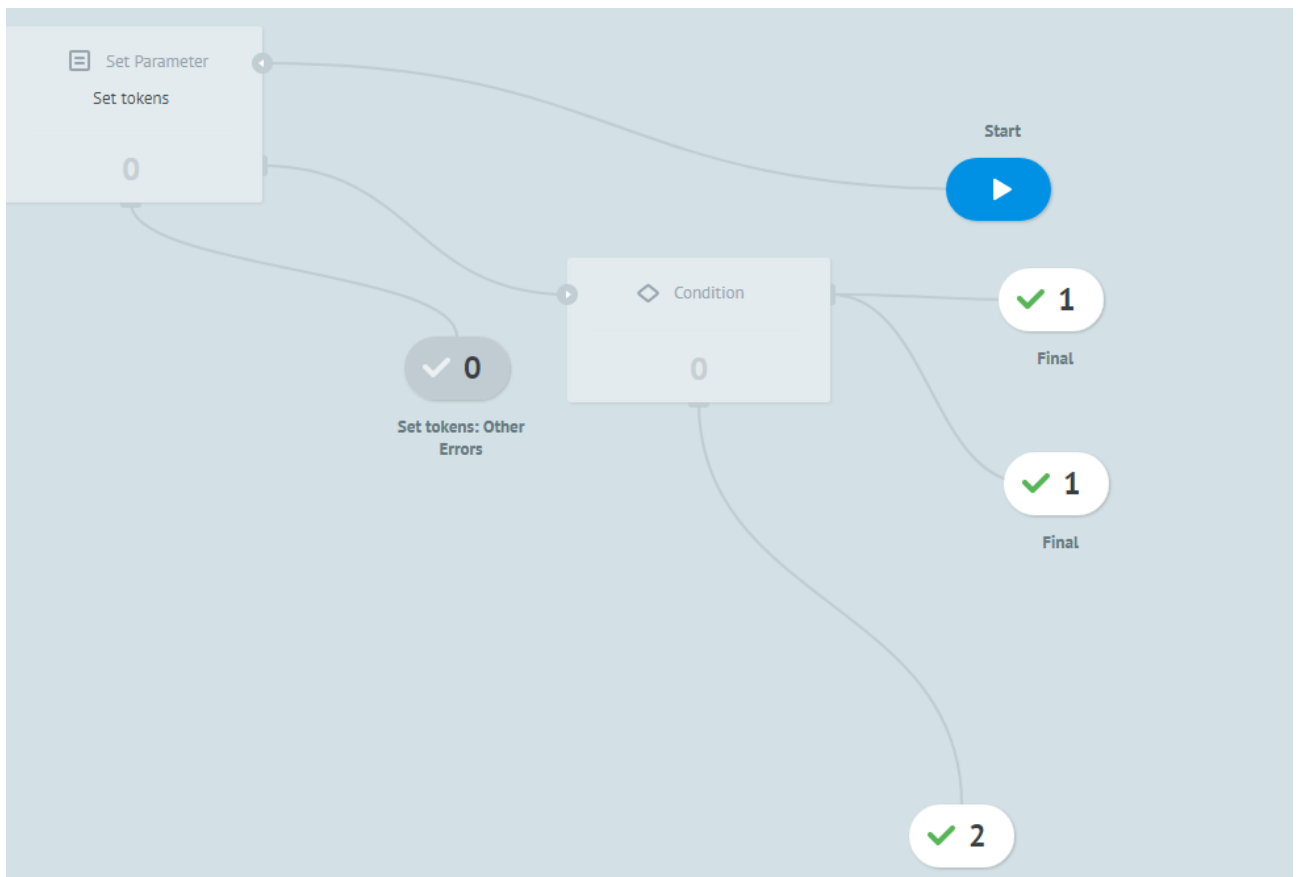


Рисунок 12 – Результат роботи умов

### Тестування REST API.

Для цього завантажимо розширення «ARC cookieexchange» <https://chrome.google.com/webstore/search/ARC%20REST?hl=uk>, або програму «Postman» <https://www.getpostman.com/apps>.

Заходимо на сторінку «TelegramBotApi» де є все необхідне для роботи з API <https://core.telegram.org/bots/api>. Для запитів до бота використовується наступний URL [https://api.telegram.org/bot<token>/METHOD\\_NAME](https://api.telegram.org/bot<token>/METHOD_NAME), де <token>– це наш токен,



який нам надав BotFather у повідомленні, а **METHOD\_NAME** – це ім'я запиту. Найпростіший запит GET запит – getMe. Результат запиту наведений на рис. 14.

<https://api.telegram.org/bot<token>/getMe>

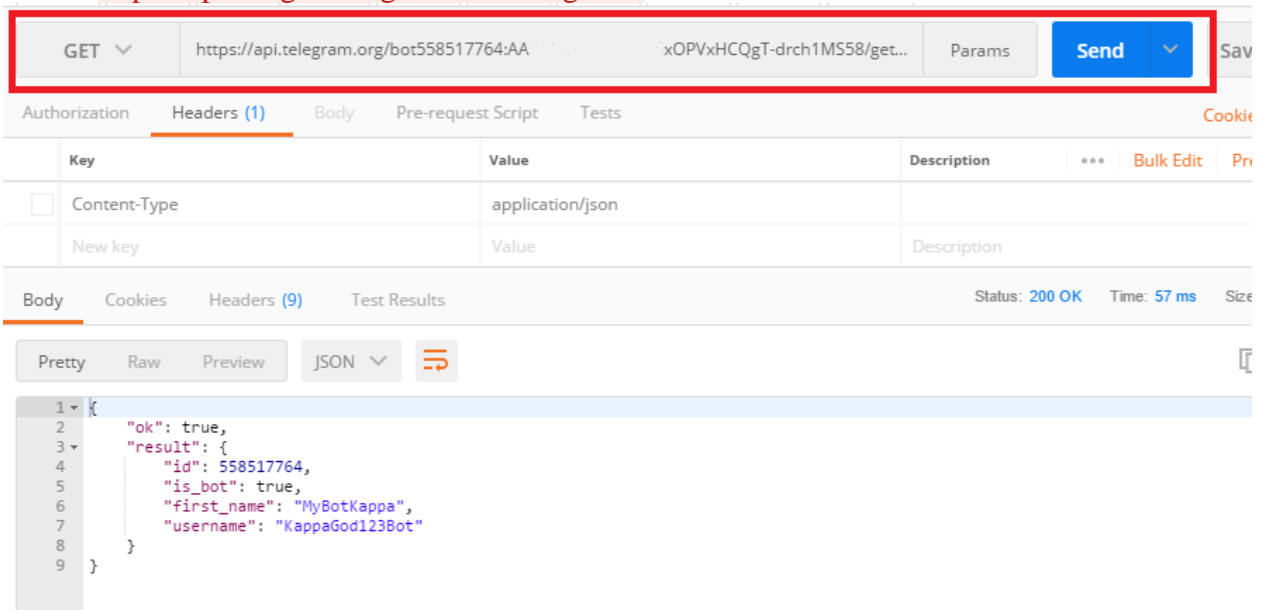


Рисунок 13 – Результат запиту getMe

Відправимо повідомлення запитом нашому боту. Замість getMe, sendMessage, обираємо метод POST. Для цього запиту, потрібні поля «text» та «chat\_id». У chat\_id записуємо id, яке показано у Corezoid. Для цього перейдіть у розділ View, та натисніть на будь який з Final, де є інформація про ваші повідомлення рис. 14.

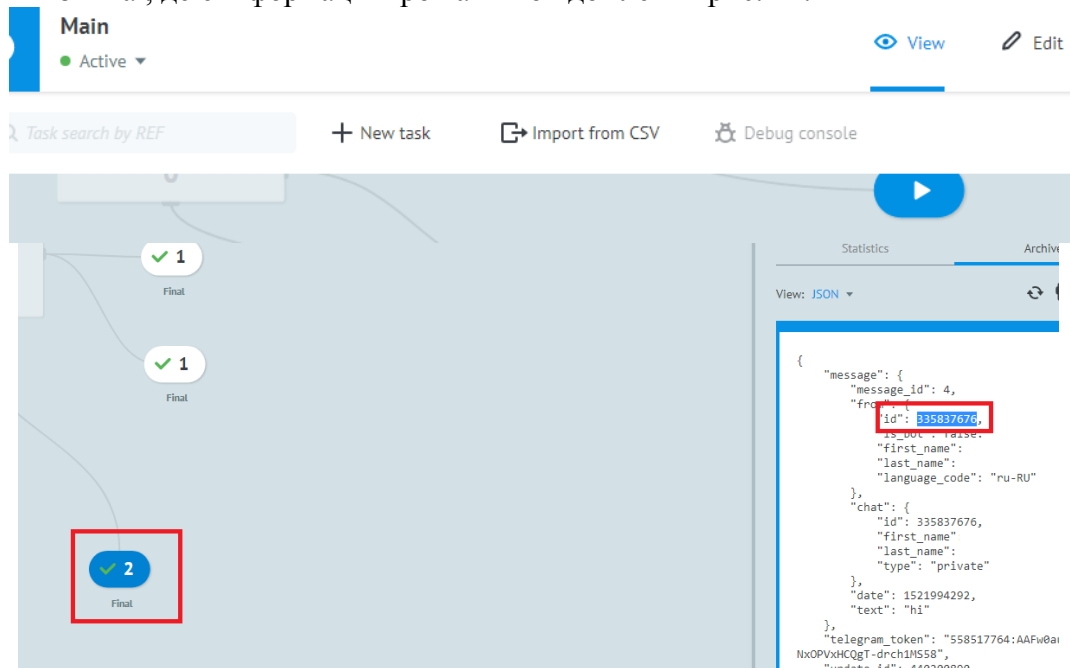


Рисунок 14 – id для запиту

Обираємо тип для запиту raw JSON(application/json),

```
{
  "text": "Hi !",
  "chat_id": 335837676 (у вас інше id)
}
```

Після вдалого запиту, у вас буде результат зображений на рисунку 15.

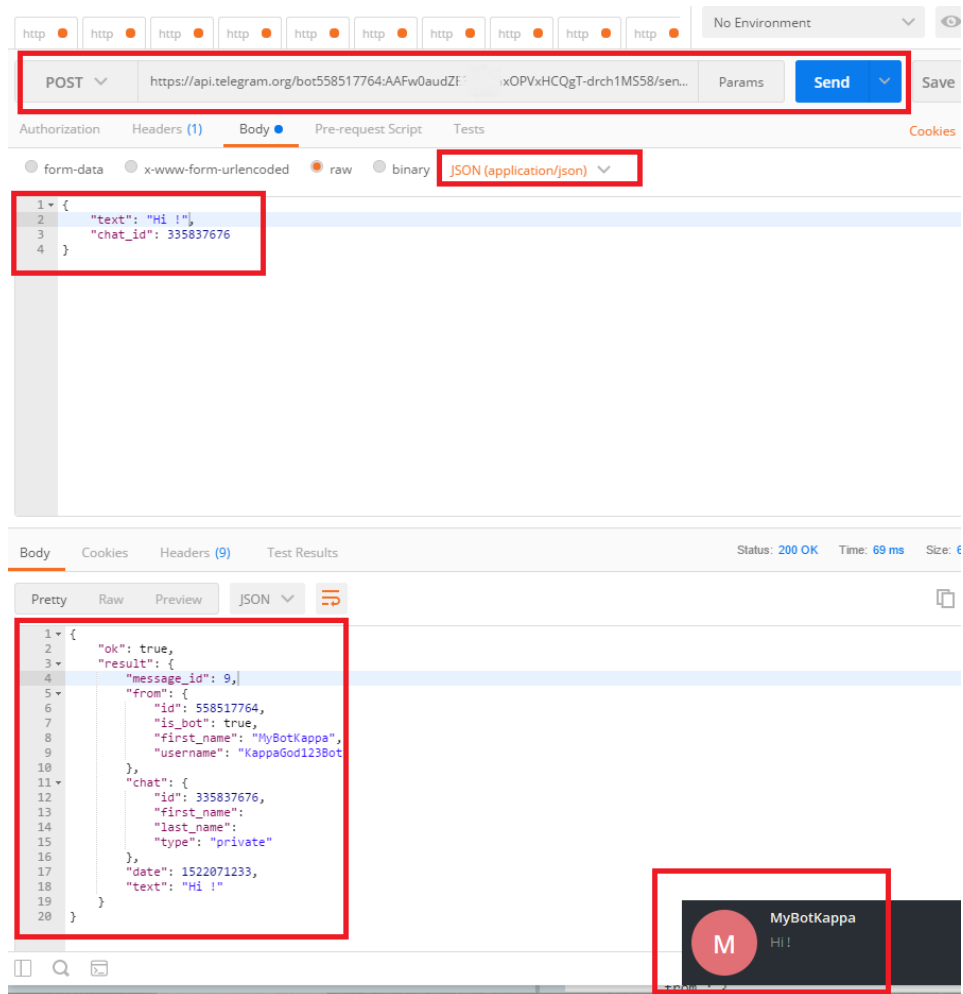


Рисунок 15 – Результат вдалого запиту POST sendMessage

Додаємо модуль ApiCall до одного з Final нашої умови рис 16.

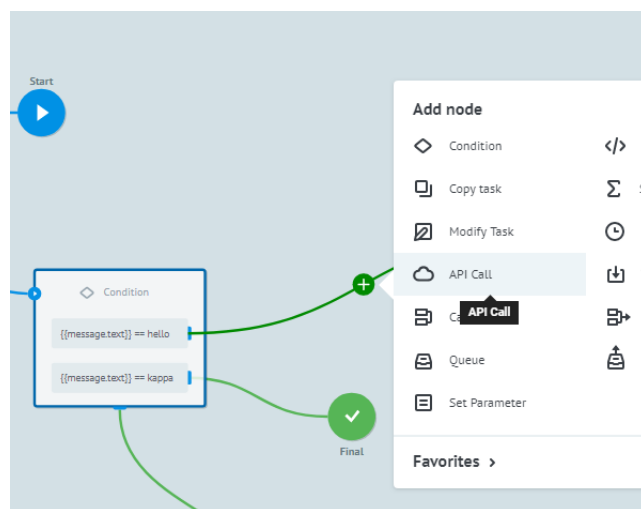


Рисунок 16 – Додавання ApiCall

Після чого буде додано модуль з додатковими елементами. Вони нам не потрібні окрім Final та Error який ми з'єднуємо перетягнувши зв'язок від «+» до кружечка Error. Після видалення непотрібних елементів ApiCall буде виглядати як на рисунку 17.

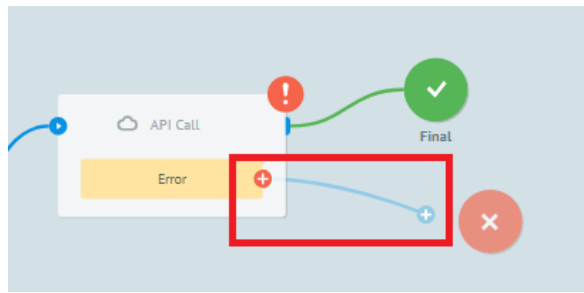


Рисунок 17 – ApiCall

Додаємо змінні до процесу натиснувши на блок «Set Parameter» і записуємо у додані «key-value» рис. 18

**chat\_id{{message.chat.id}}**  
**in\_text{{message.text}}**

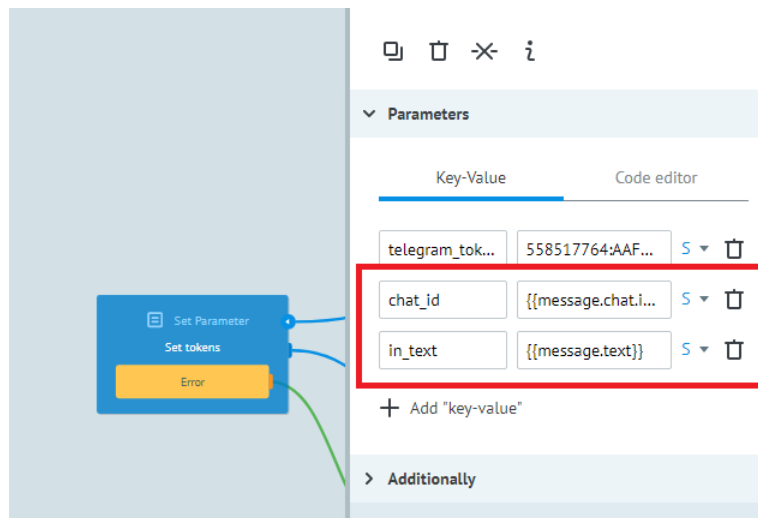


Рисунок 18 – Додавання змінних процес

Додані зміни використовуємо у блоці «Api Call»рис. 19.

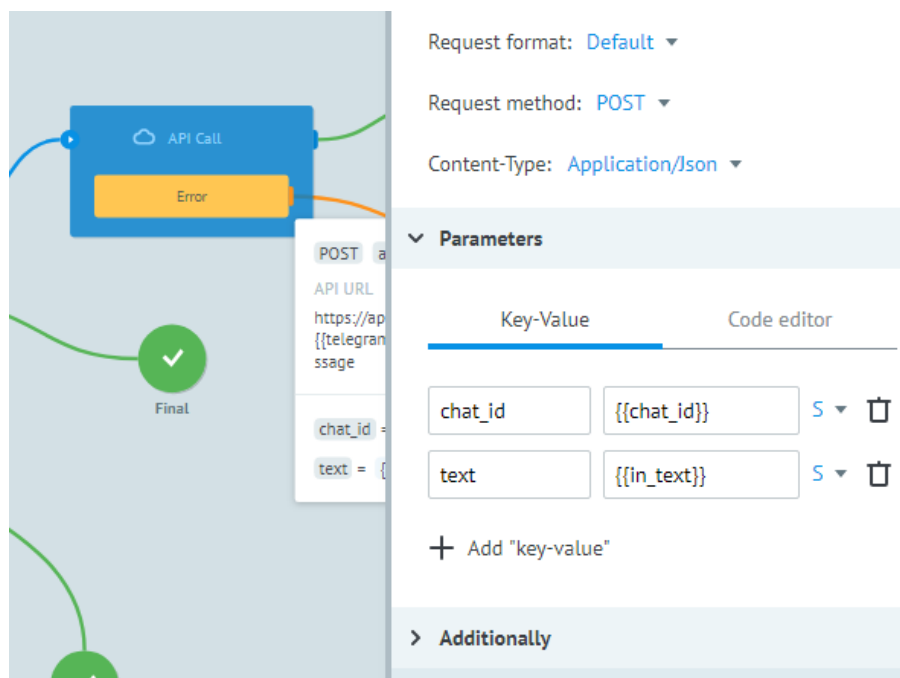


Рисунок 19 – Додавання параметрів у блок «Api Call»

Якщо все було зроблено вірно, то бот відповість вам, якщо ви напишете слово, на якому у вас стоїть умова. У даному випадку це слово «hello» рис. 20.

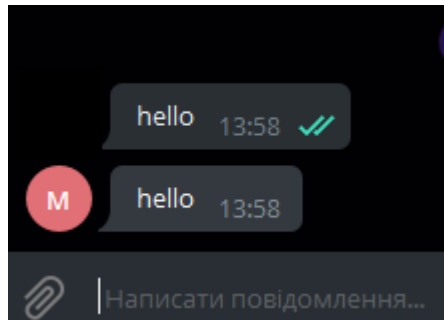


Рисунок 20 – Відповідь бота

Скопіюємо створену умов комбінацією Ctrl+C Ctrl+V. Додамо копію умови на нашу «Final» перетягнувши на гілку до «Final» так, щоб курсор спів пав з гілкою. Скопійований блок стане прозорим рис. 21 – 23.

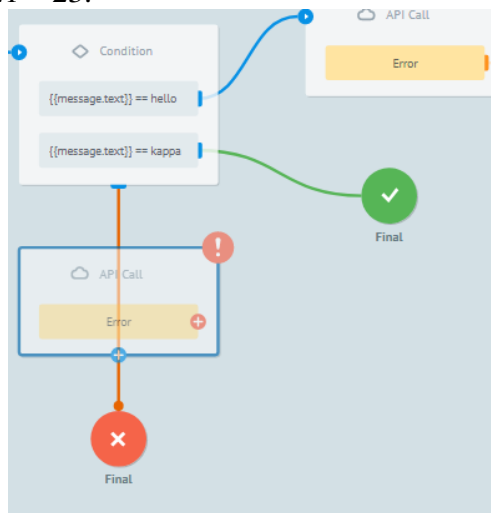


Рисунок 21 – Додавання умови

На вільний червоний плюс, додаємо вузол «End error» натиснувши на нього та вибравши зі списку.

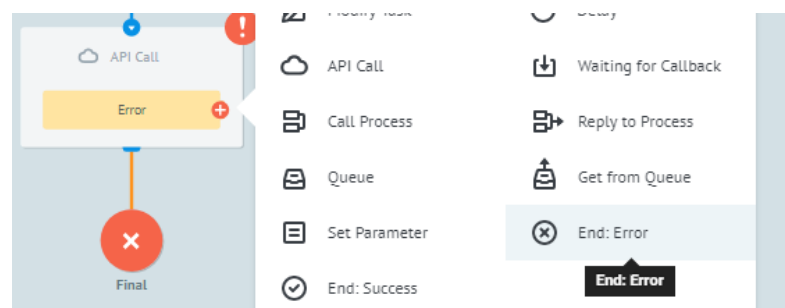


Рисунок 22 – додавання вузла

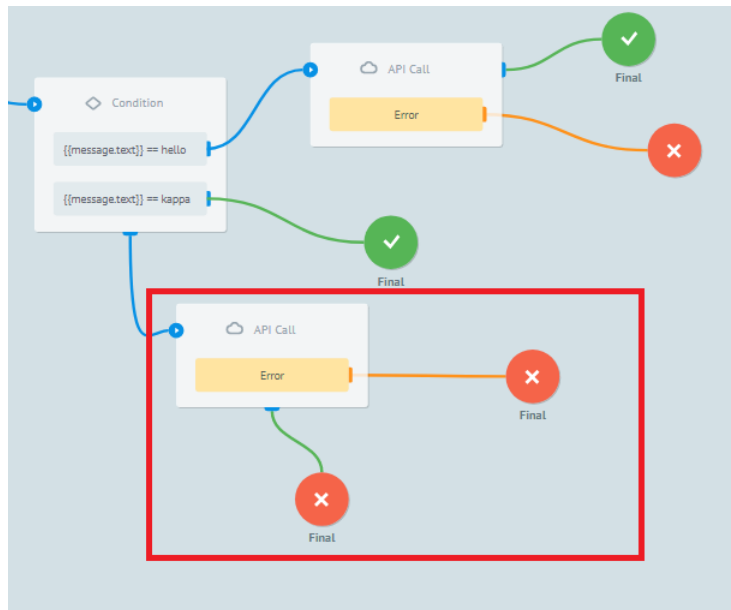


Рисунок 23 – Додана умова

Тепер написавши будь що, він буде відповідати теж саме рис 24.

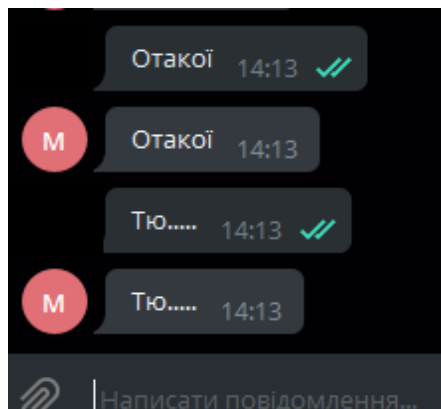


Рисунок 24 – Відповідь бота на написані повідомлення

### ДОДАВАННЯ ПОГОДИ У БОТ

Наступний запит виведе погоду, яка є зараз:

`query.yahooapis.com/v1/public/yql?q=select item.condition from weather.forecast where woeid in (select woeid from geo.places(1) where text='vinnitsa') and u='c'&format=json`

Результат цього запиту буде json файл рис 25.

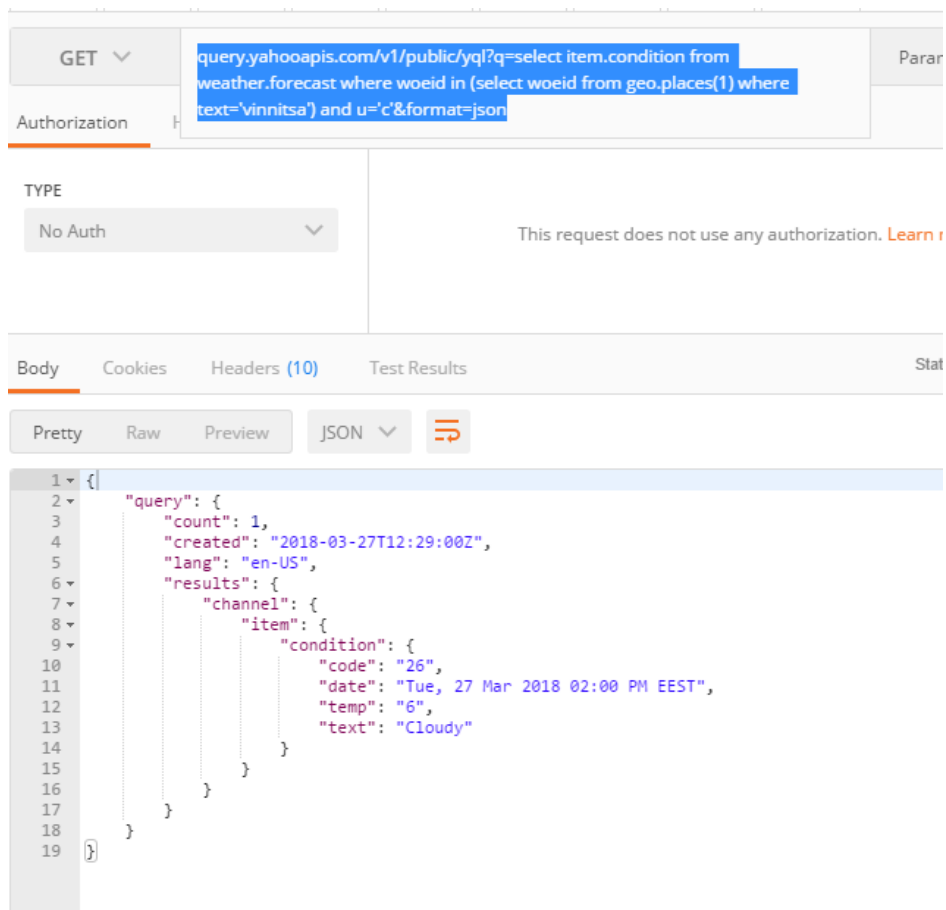


Рисунок 25 – Отриманий JSON файл

Для реалізації цього у боті, створюємо новий процес. Щоб перейти у структуру бота, потрібно натиснути логотип Corezoid рис. 25. Після чого додамо новий процес рис. 26 та назвемо його «Weather»

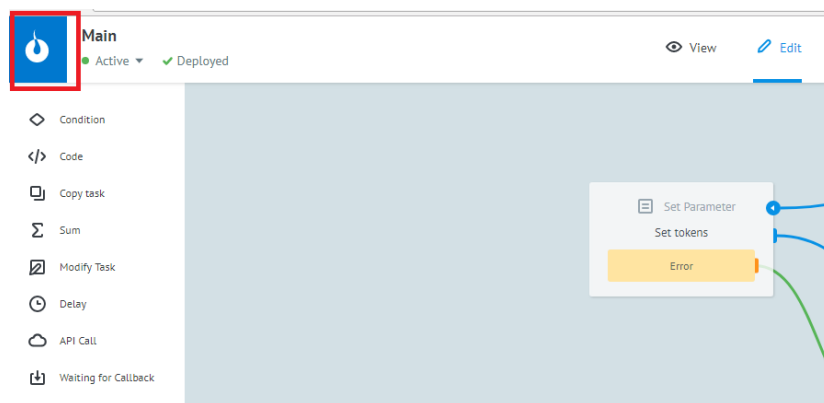


Рисунок 25 – кнопка для переходу у структуру бота

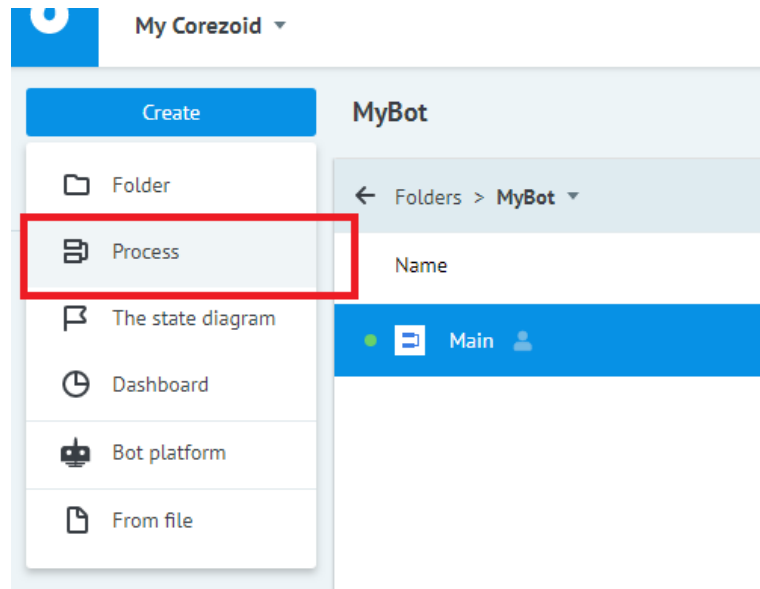


Рисунок 26 – Додавання процесу

Під'єднання нового процесу до головного.  
Заходимо у редактор процесу «Main» і додаємо нову умову рис. 27.

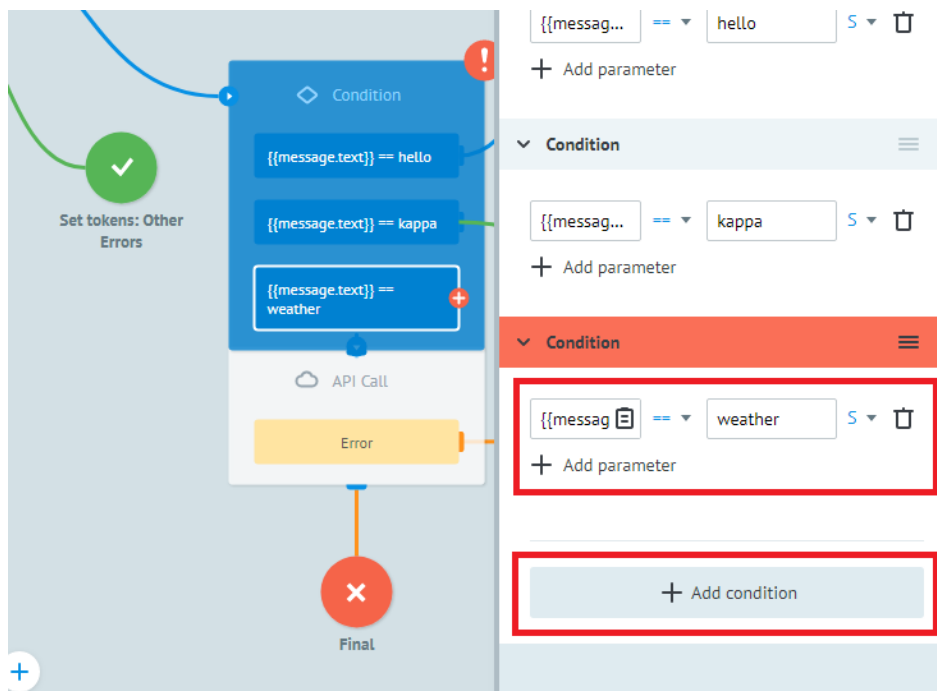


Рисунок 27 – Додавання нової мови

{{message.text}} weather

І на нову умову, додаємо «Call Process» рис. 28.

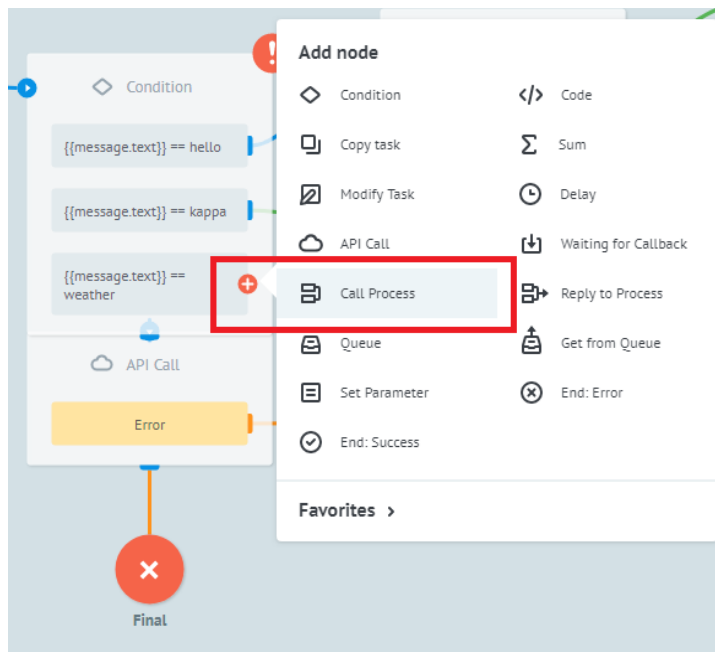


Рисунок 28 – Додавання «Call Process»

Далі потрібно вибрати процес. Для цього потрібно натиснути на кнопку у полі для назви процесу рис. 29 та вибрати потрібний процес.

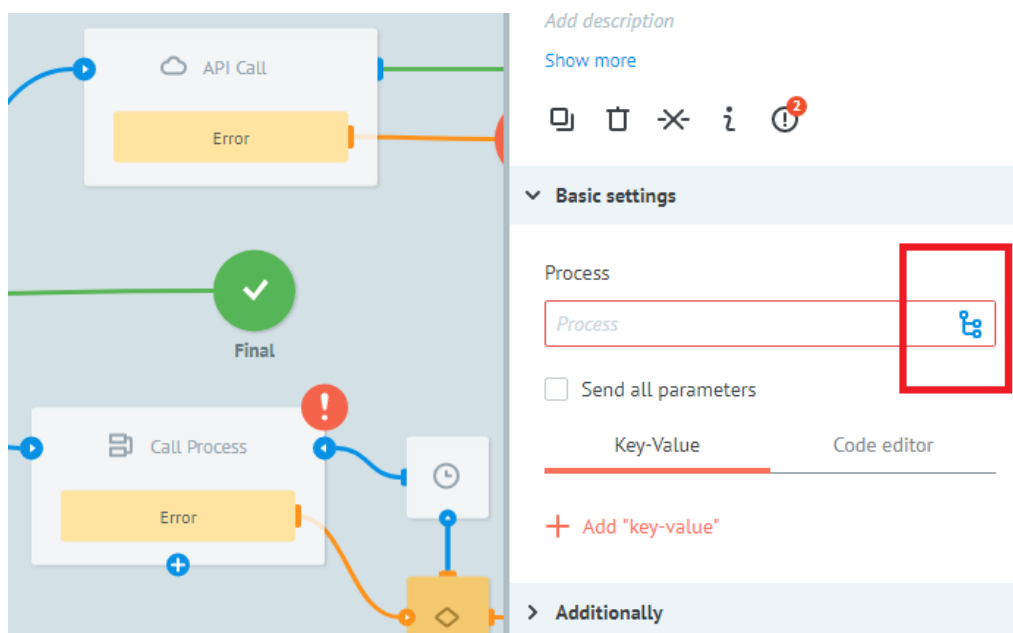


Рисунок 29 – кнопка для вибору процесу для виклику

Видаляємо непотрібні вузли, додаємо параметри блока, які він буде приймати рис. 30.

```

telegram_token{    {telegram_token}}
chat_id            {{chat_id}}
in_text           {{in_text}}

```



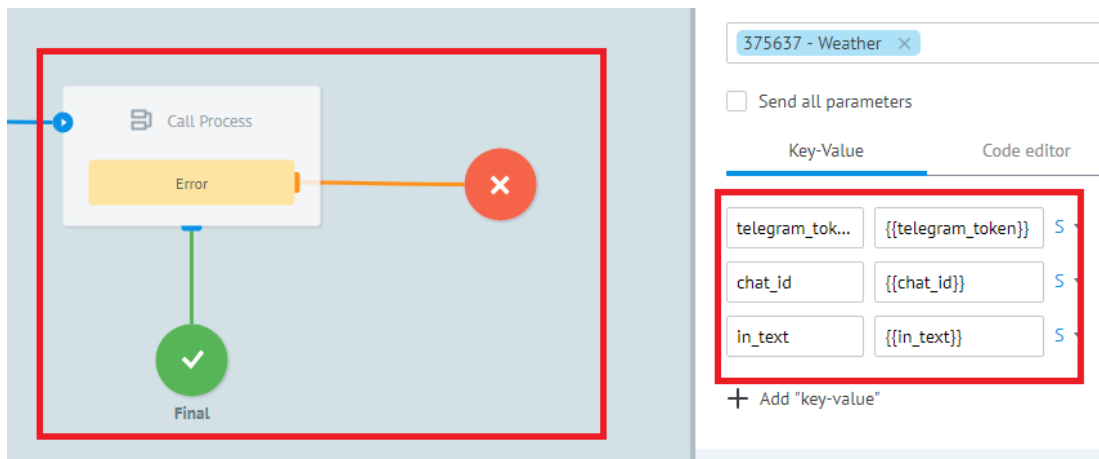


Рисунок 30 – Параметри блоку «Call Process»

Знову повертаємось до нашого нового процесу та додаємо блок «Api Call»

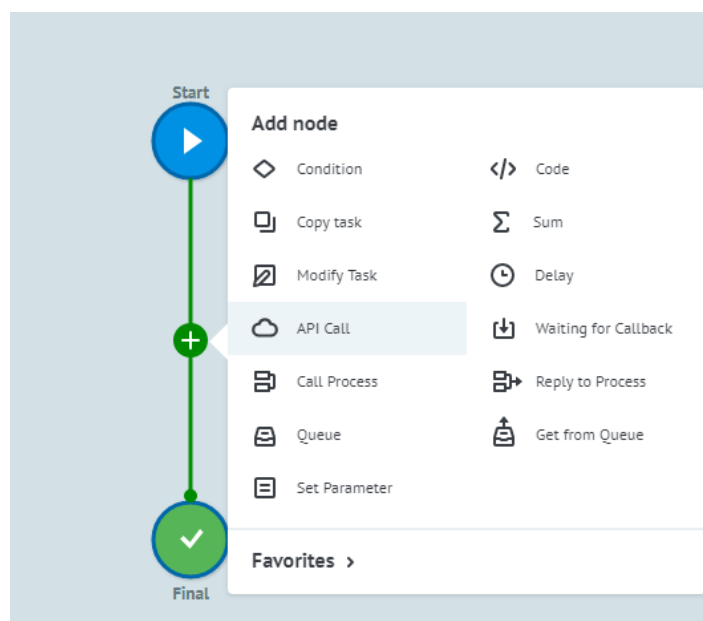


Рисунок 31 – Додавання блоку «Api Call»

Налаштовуємо «Api Call». Налаштований «Api Call» зображений на рис. 32.

**url:** <https://query.yahooapis.com/v1/public/yql>

**q:**selectitem.conditionfromweather.forecastwherewoeidin (selectwoeidfromgeo.places(1) wheretext='vinnitsa') and u='c'

**format:** json



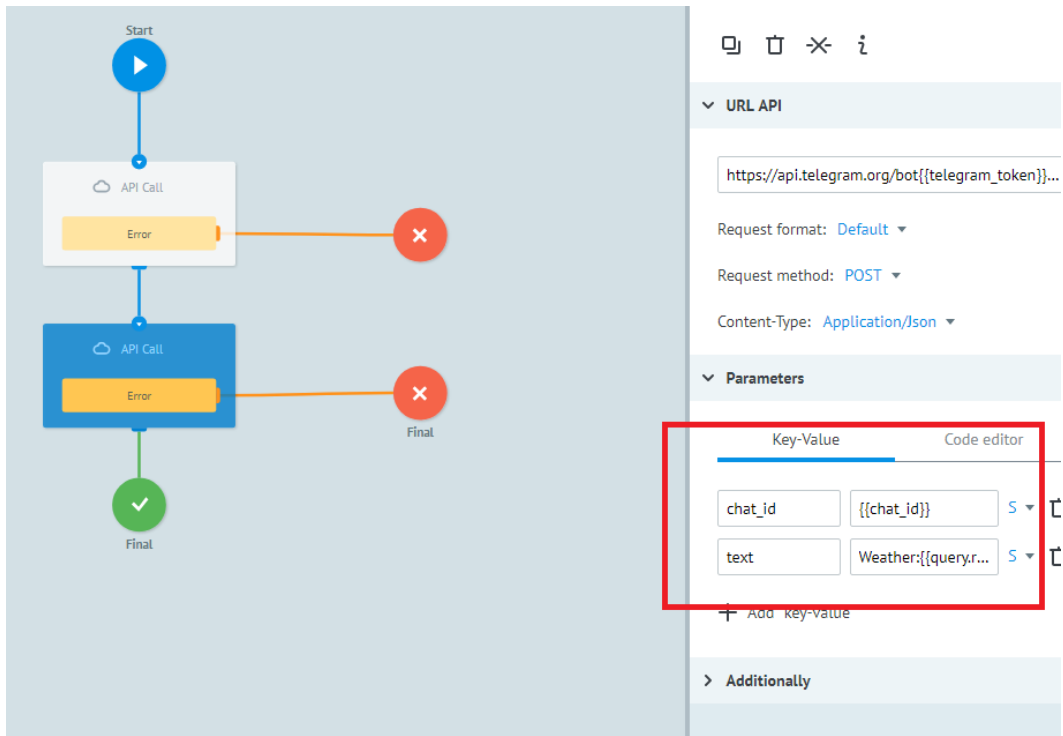


Рисунок 34 – Налаштування процесу «Weather»

Якщо все було виконано правильно бот на спеціальне повідомлення, дасть наступну відповідь рис. 35.

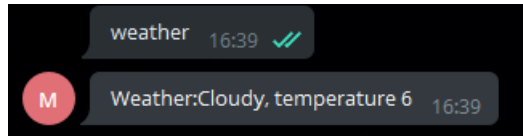


Рисунок 35 – повідомлення бота про погоду

## Лабораторна робота № 11. Створення календаря та робота зі звітами

Для створення календаря необхідно створити нову таблицю CALENDAR у SQL Workshop як показано на рисунку 1.

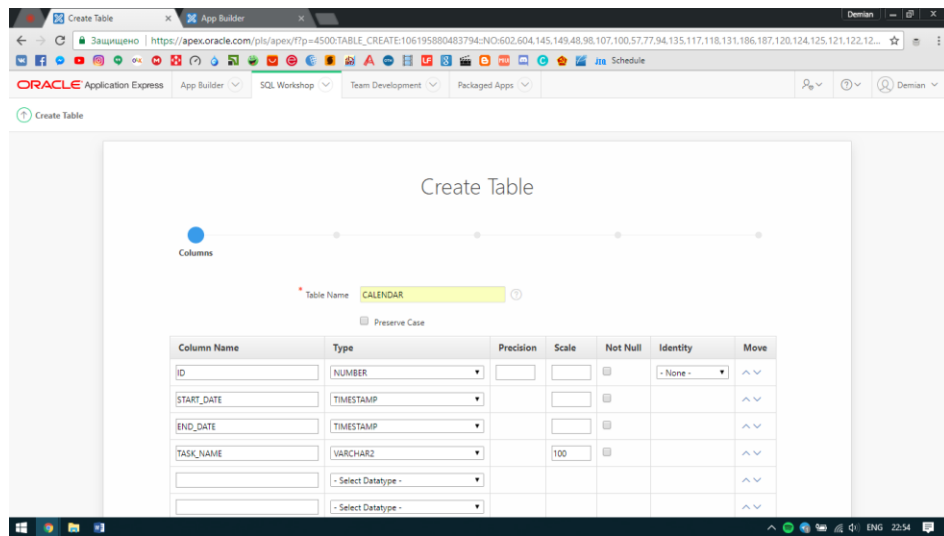


Рисунок 1 – Створення нової таблиці CALENDAR

Встановити ключовим полем (Primary Key) стовпець ID (рис. 2) та завершити процес створення таблиці нічого не змінюючи в наступних пунктах.

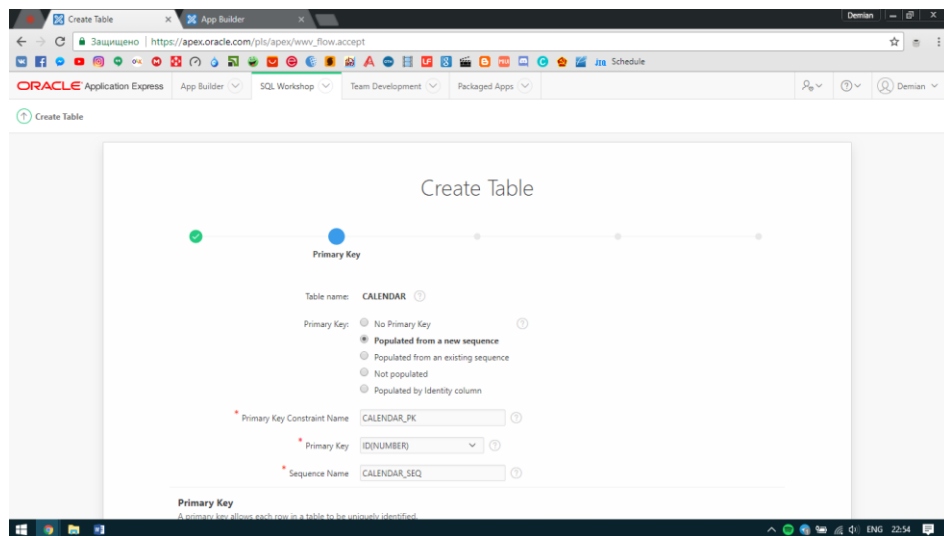


Рисунок 2 – Встановлення стовпця ID як ключового поля

Для створення календаря можна скористатися готовою сторінкою Calendar в меню додавання нових сторінок (рис. 3).

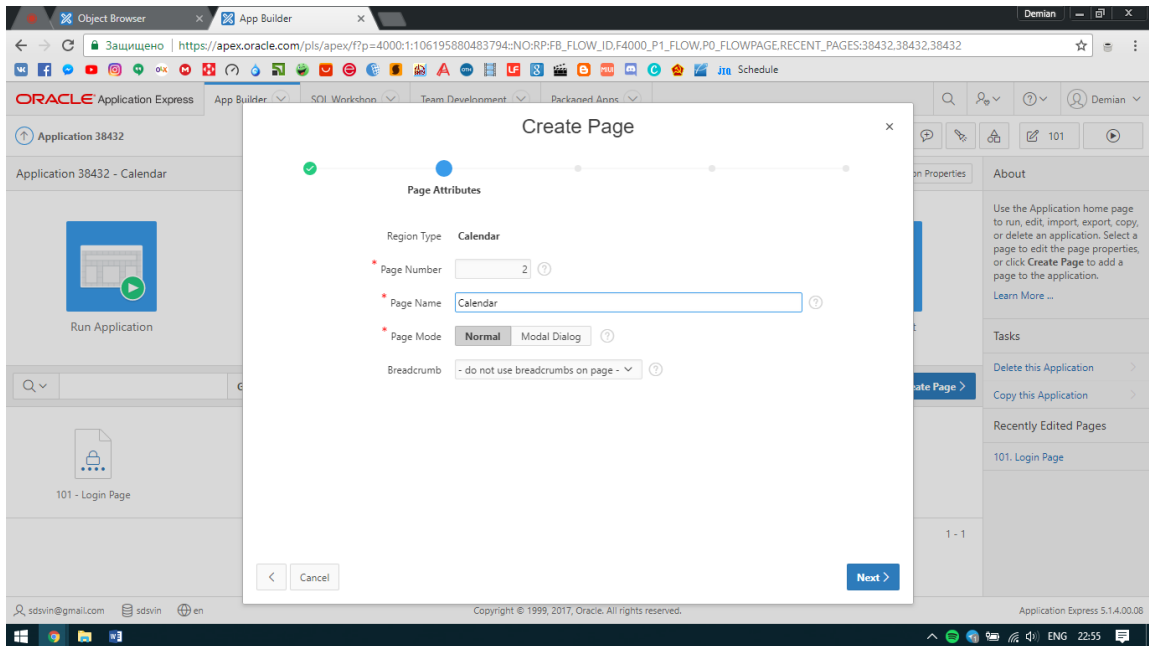


Рисунок 3 – Створення сторінки Calendar

При створенні сторінки на вкладці Source необхідно обрати створену таблицю та всі її стовпці перемістити у праву колонку (рис. 4).

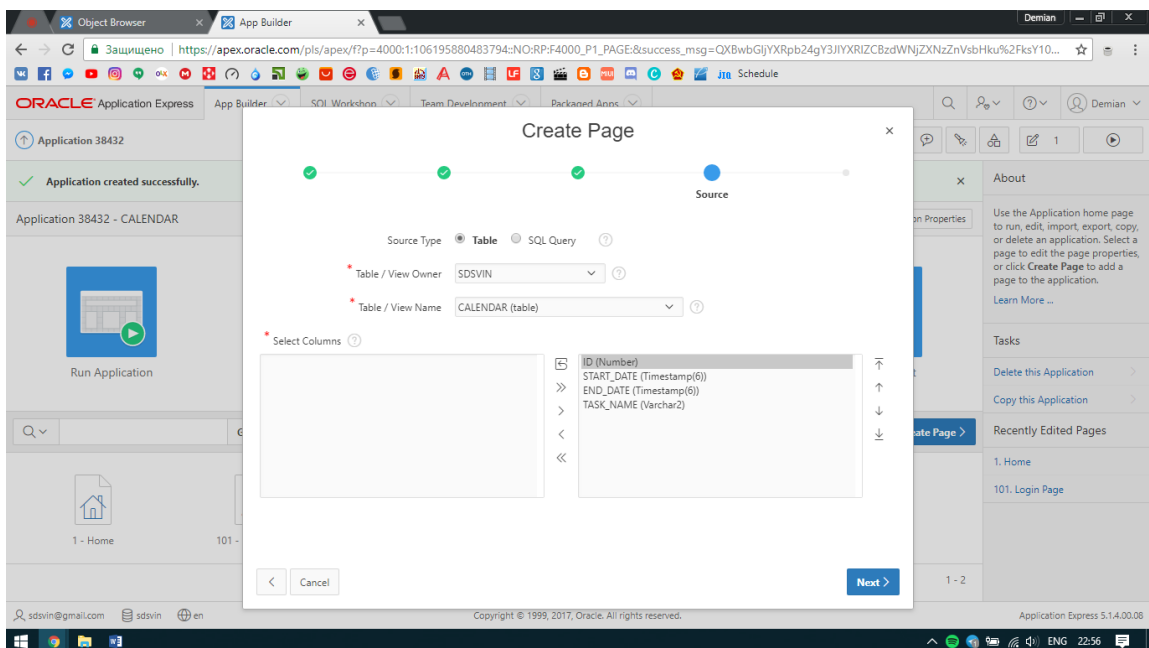


Рисунок 4 – Вибір таблиці при створенні сторінки з календарем  
На вкладці Settings необхідно встановити параметри як показано на рисунку 5.

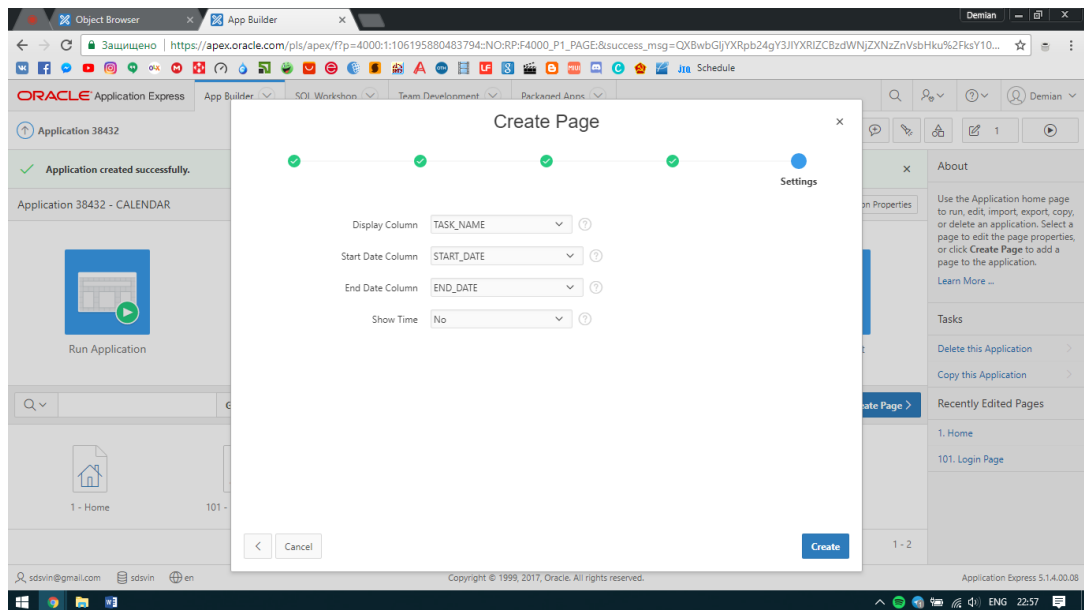


Рисунок 5 – Встановлення параметрів відображення стовпців

На створеній сторінці в атрибутах регіону Calendar необхідно встановити значення Primary Key Column на ID (рис. 6).

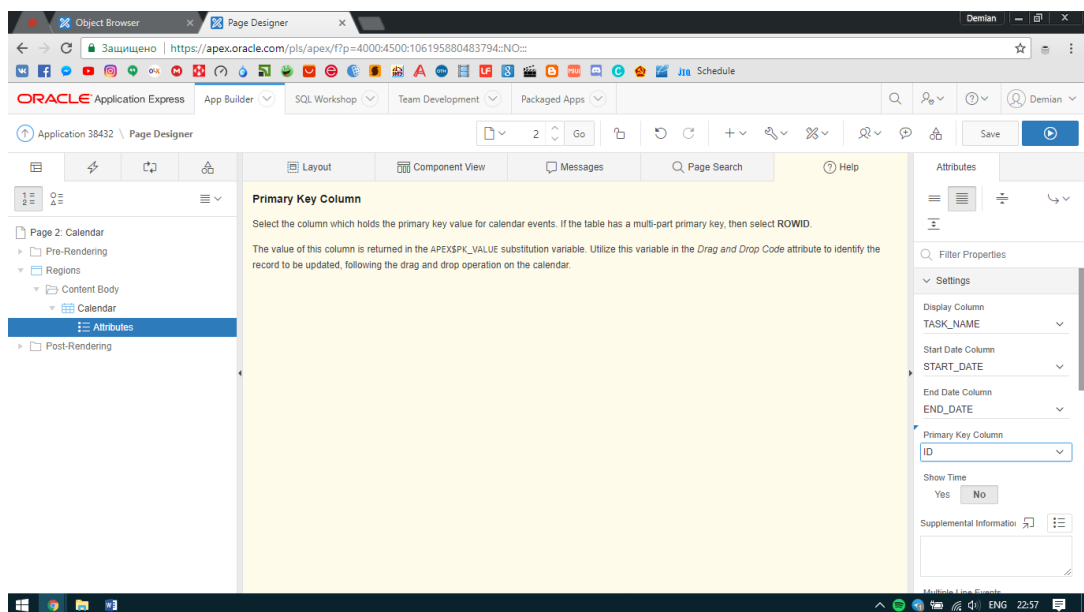


Рисунок 6 – Встановлення значення Primary Key Column на ID

Для відображення інформації в календарі необхідно створити ще дві сторінки Report та Modal типу Report with Form, що створюються одночасно (рис. 7)

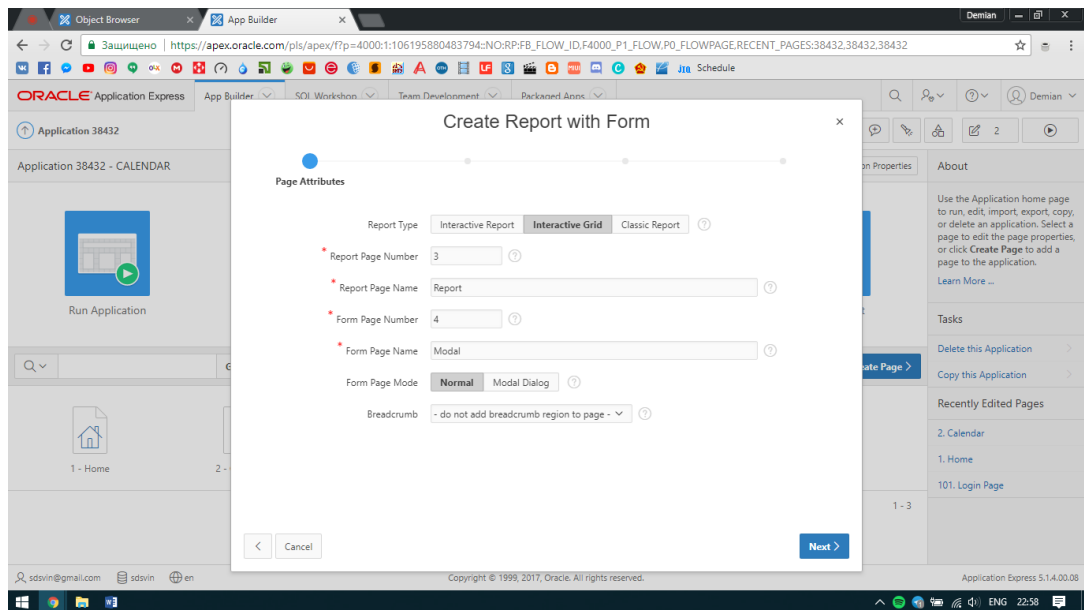


Рисунок 7 – Створення сторінок Report with Form

На вкладці Data Source необхідно обрати створену таблицю Calendar та перемістити усі стовпці у праву колонку (рис. 8).

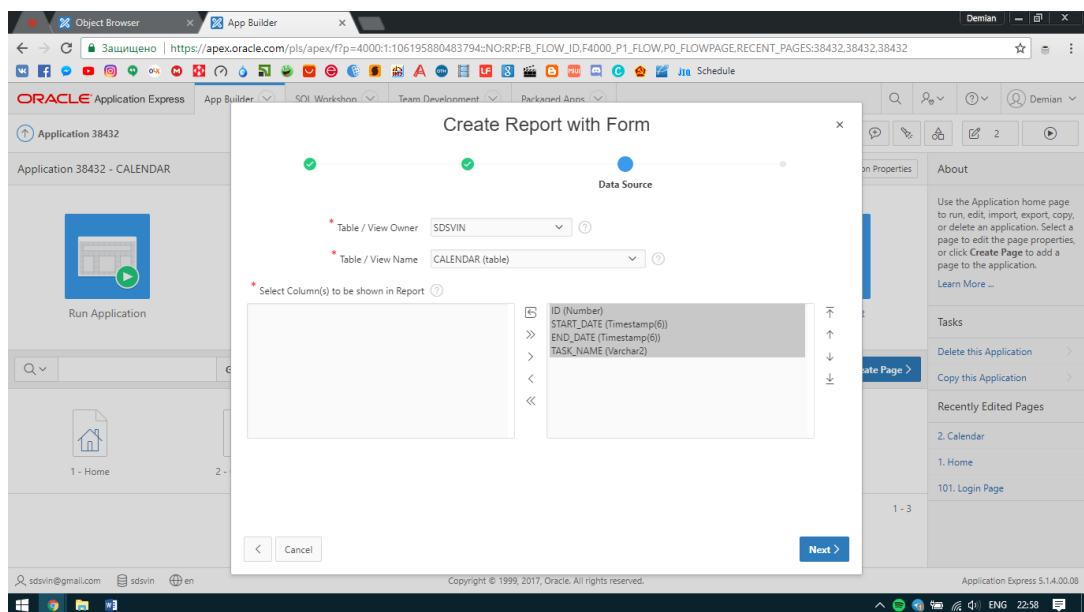


Рисунок 8 – Вибір створеної таблиці на сторінці Report with Form

На вкладці Form Page необхідно обрати створену таблицю Calendar та перемістити усі стовпці у праву колонку, обравши стовпець ID у полі Primary Key Column 1 (рис. 9).

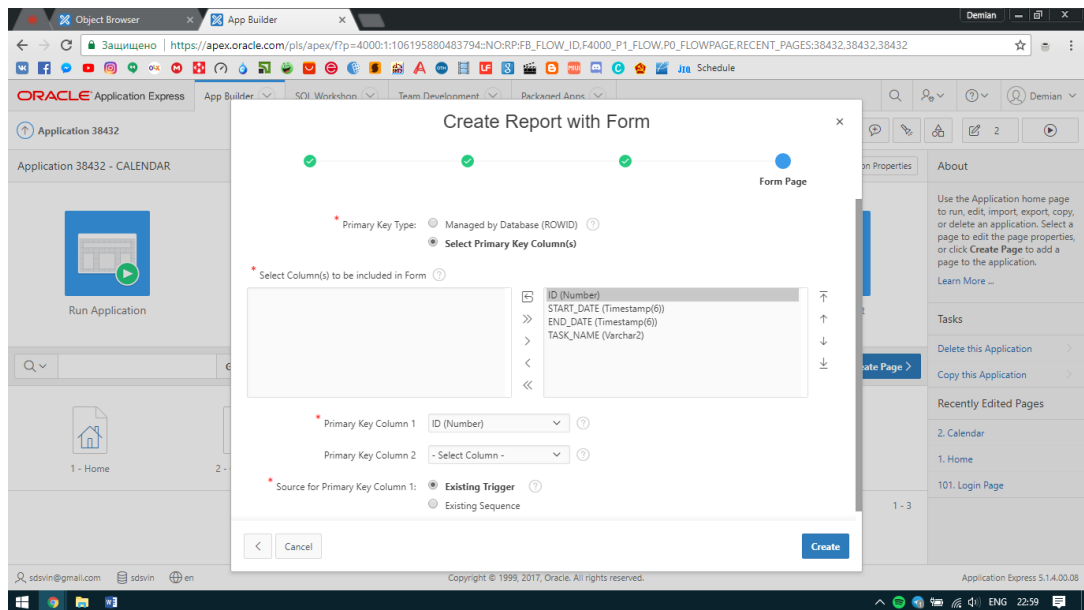


Рисунок 9 – Встановлення Primary Key Column

На сторінці Calendar в розділі Create Link необхідно створити посилання на створену форму Modal, яка буде відкриватися для редагування значення комірки (комірок) в календарі (рис. 10).

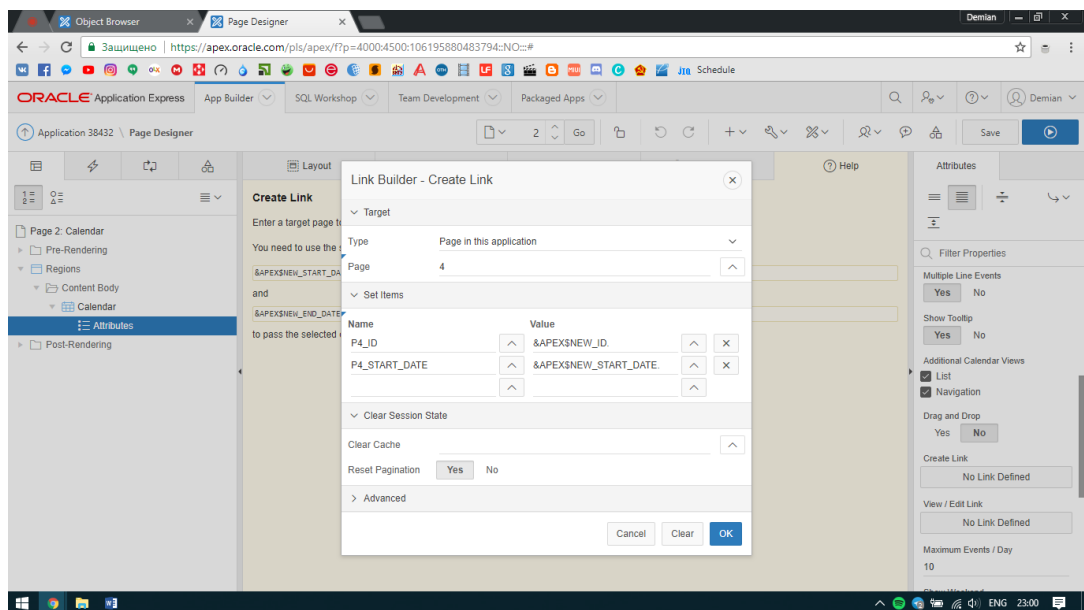


Рисунок 10 – Створення посилання на форму Modal

На сторінці Calendar в розділі View/Edit Link необхідно створити посилання на створений звіт Report, який буде відкриватися після внесення значення у комірку (комірки) в календарі (рис. 11).



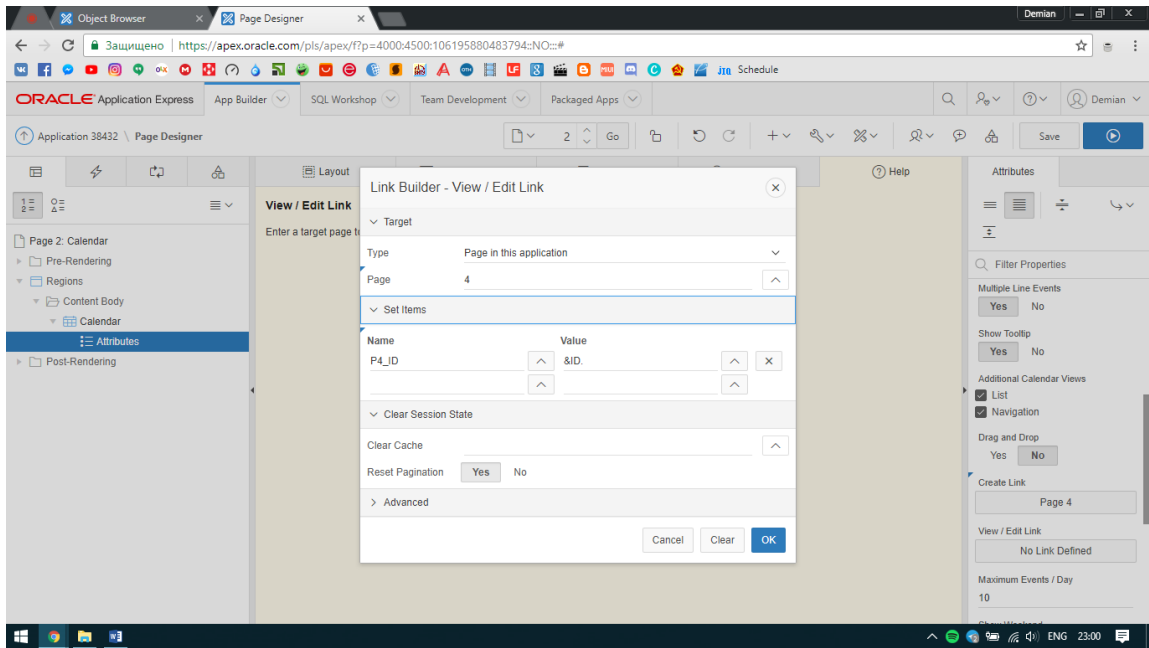


Рисунок 11 – Створення посилання на звіт Report

Вигляд календаря показано на рисунку 12.

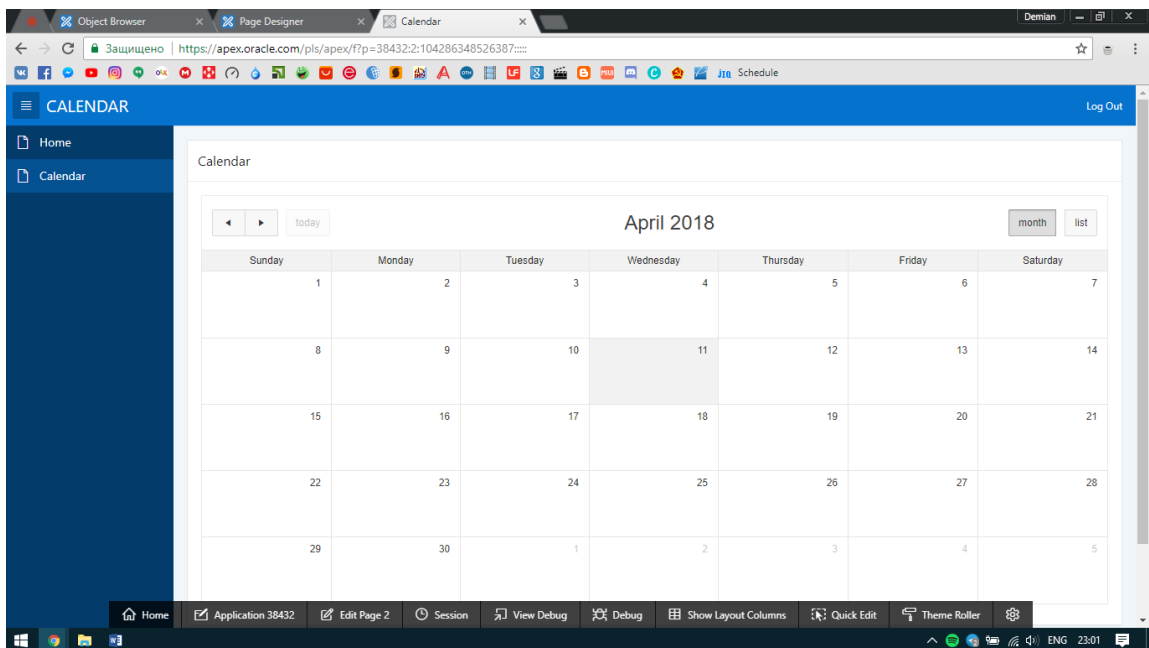


Рисунок 12 – Вигляд календаря

При натисканні на пусте місце в комірці відкривається форма для введення інформації в календар (рис. 13).

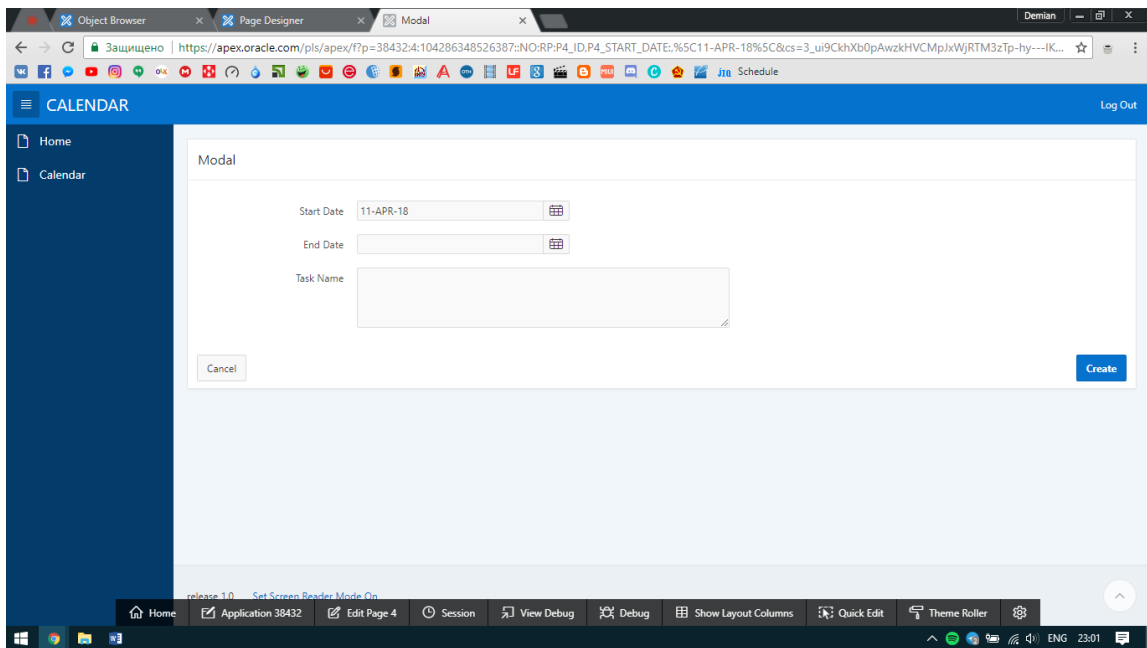


Рисунок 13 – Форма для введення інформації в календар

Після натискання кнопки Create відкривається звіт та повідомлення про успішний запис даних до таблиці (рис. 14).

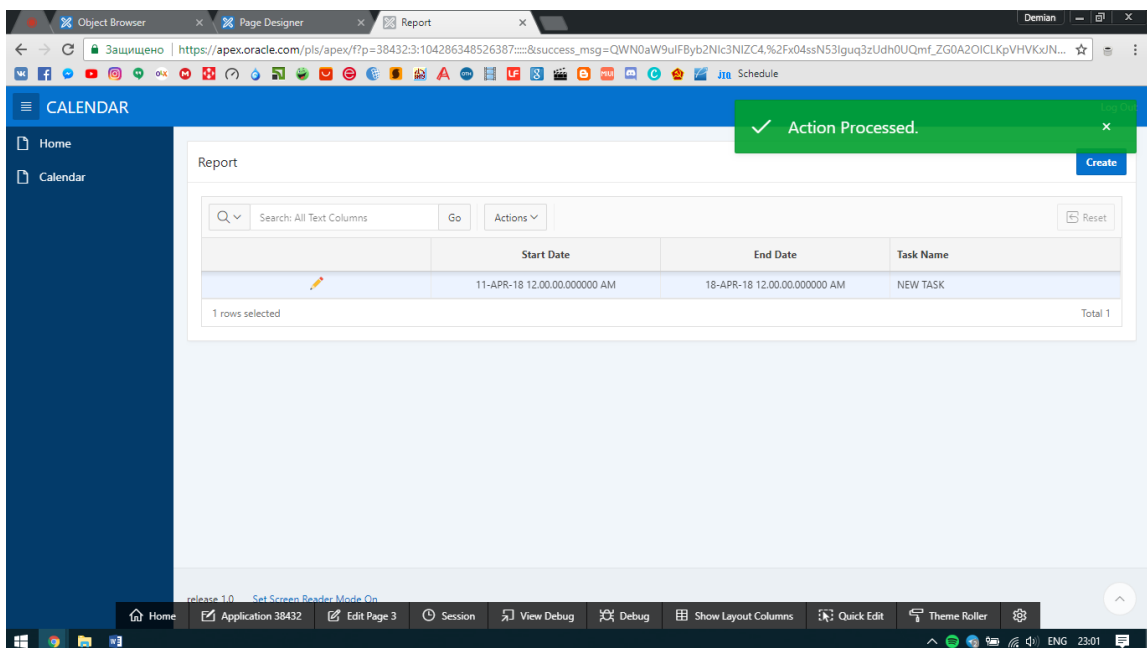


Рисунок 14 – Звіт про запис даних до таблиці  
Вигляд заповненого інформацією календаря наведено на рисунку 15.

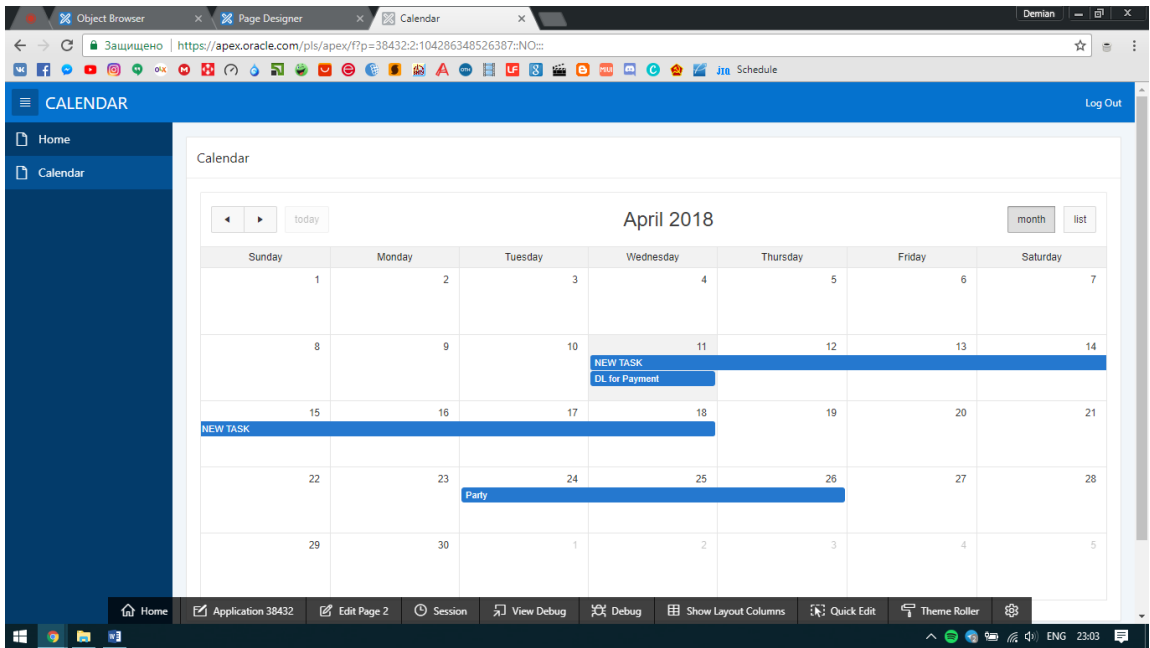


Рисунок 15 – Заповнений інформацією календар

Для редагування існуючої інформації в календарі необхідно натиснути на існуючий запис, щоб відкрилася форма для редагування (рис. 16).

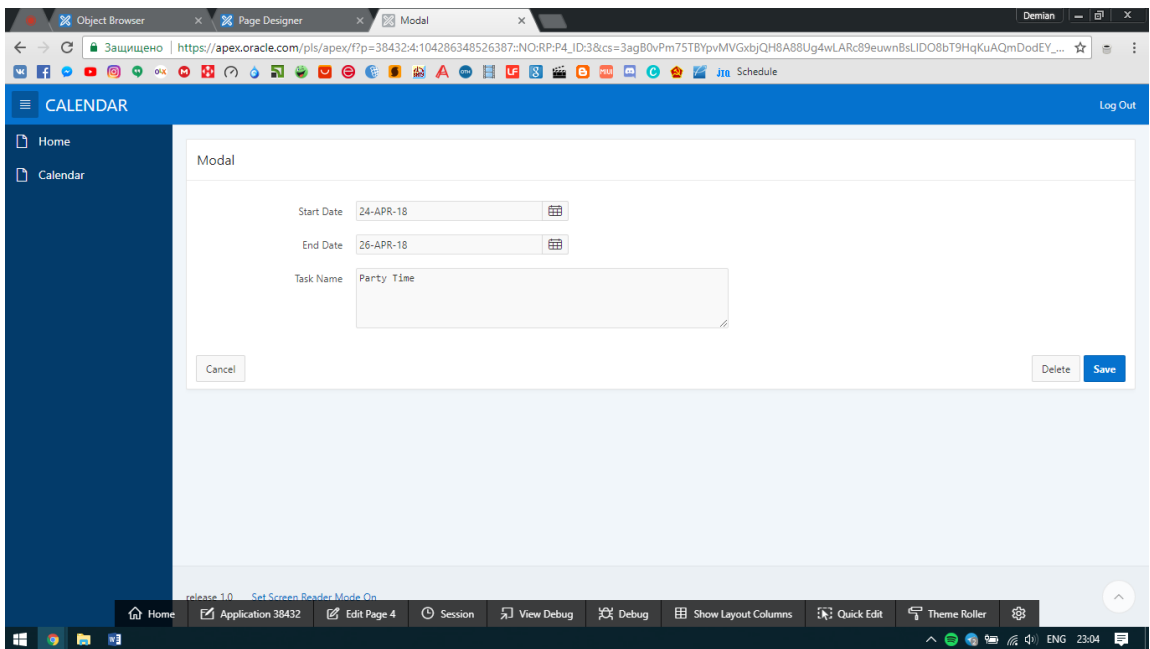


Рисунок 16 – Форма редагування записів