Юдкін Максим

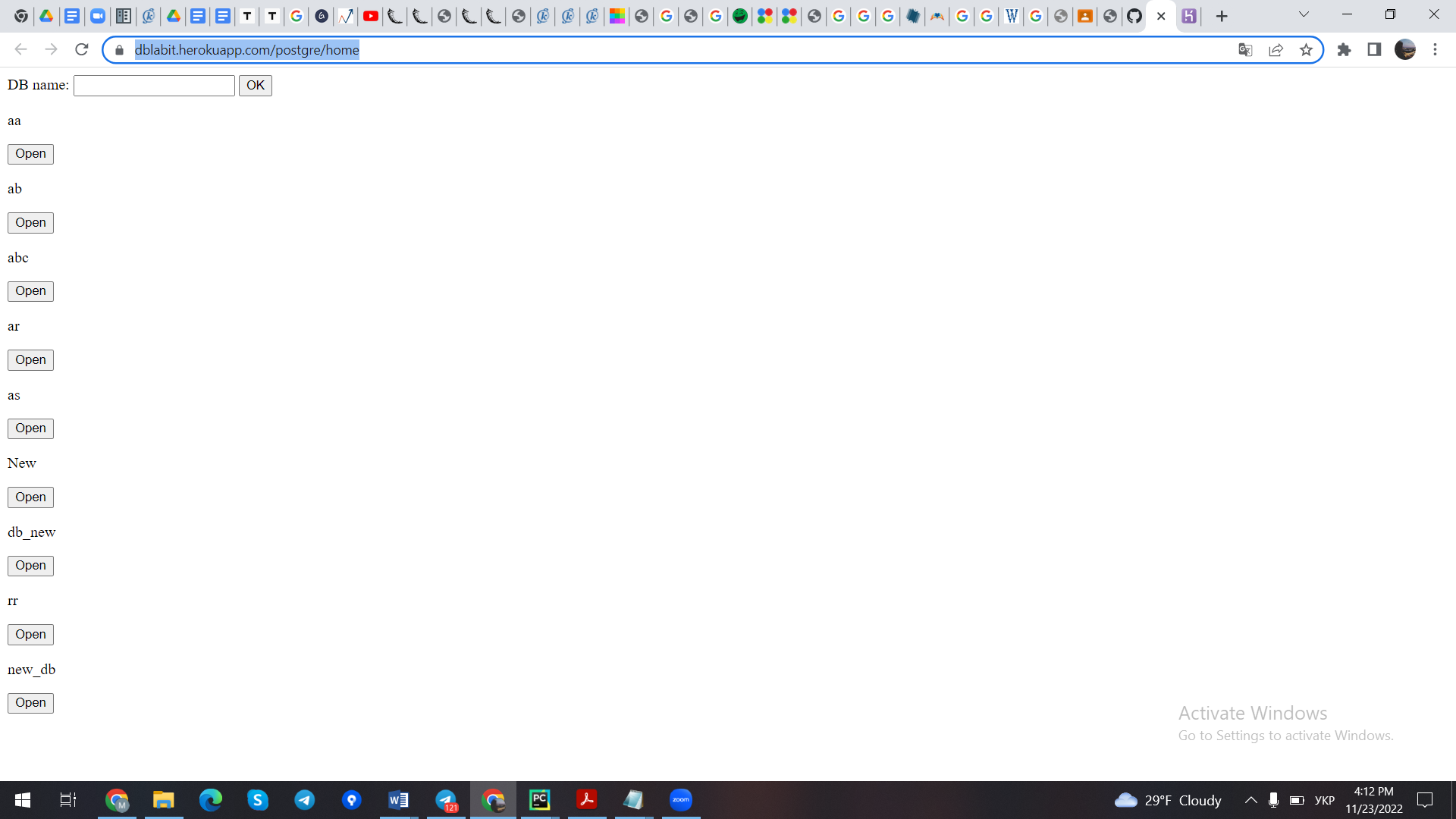
Етапи 24 та 22

Етапи 24 та 22 включають в себе створення бази даних на основі іншої бази (в моєму випадку PostgreSQL) та використовування хмарних сервісів (в моєму випадку Heroku)

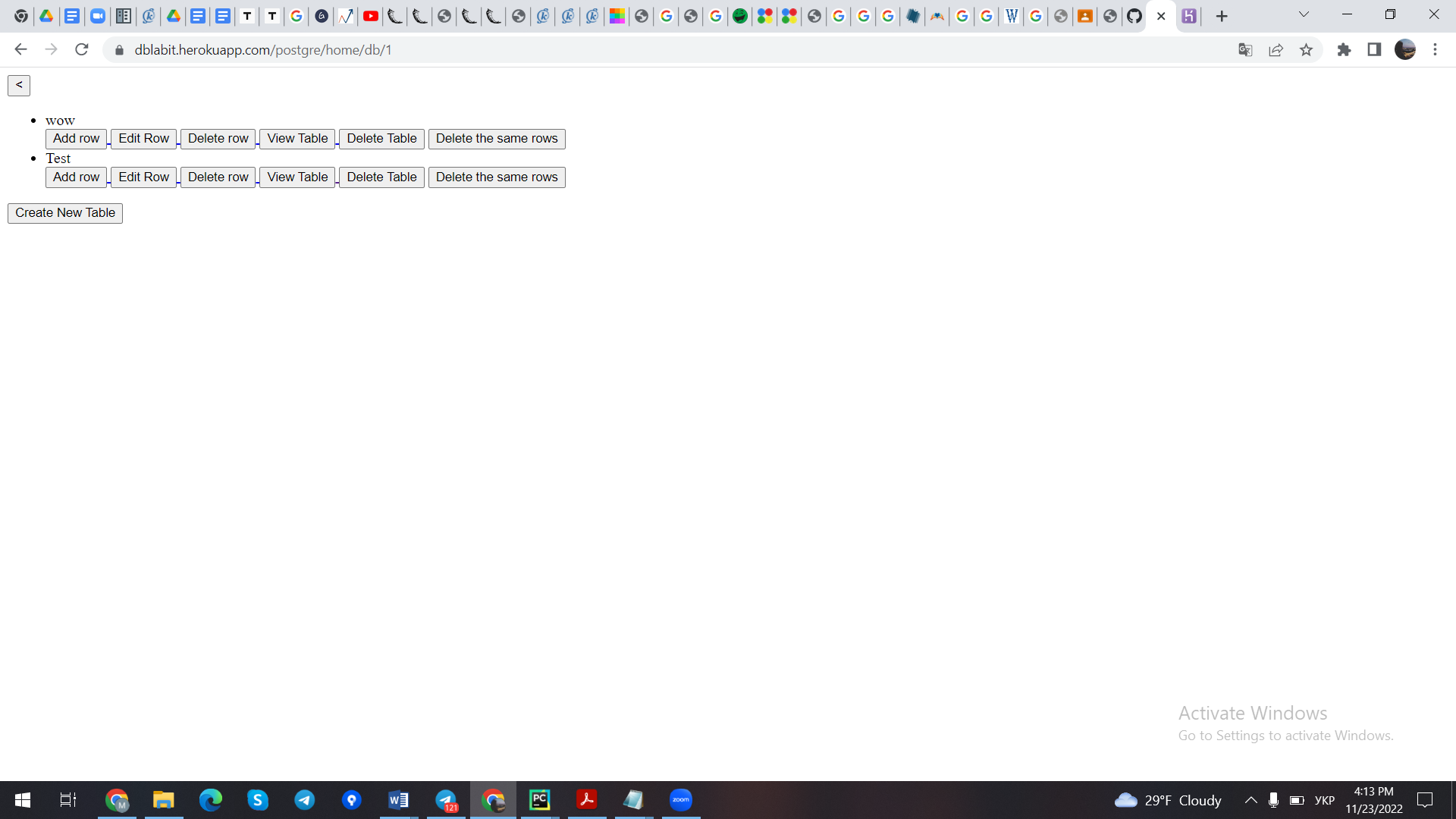
Розміщення на хероку проекту відбувається за посиланням <https://dblabit.herokuapp.com/postgre/home> з одним лише нюансом ,

Хероку обмежить використання баз данних з 28-го числа , та це буде платною функцією (але демонстрація працездатності роботи я записав на відео що прикріпив у класрумі)

А ось реалізація бд через Постгре буде виглядати так



Фронтенд майже не змінився , окрім того що не імя ідентифікує таблиці та бд а автогенеруюче айди



Білше змін застав бекенд , адже тепер звернення йде не до файлику на компі, а до бд , і робиться це через бібліотеку psycopg2 з використанням sql ось наприклад видалення однакових рядків(операція мого варіанту)

@csrf\_exempt  
def del\_same\_rows(request, db\_id, table\_id):  
 if request.method == 'DELETE':  
 conn = get\_connection()  
 cur = conn.cursor()  
 cur.execute("SELECT db\_name FROM db\_postgre\_db WHERE id= %(db\_id)s ", {"db\_id": db\_id})  
 data = cur.fetchone()  
 db\_name = data[0]  
  
 cur.execute(f"SELECT table\_name FROM db\_postgre\_table WHERE id= %(table\_id)s ", {"table\_id": table\_id})  
 data = cur.fetchone()  
 table\_name = data[0]  
 cur.execute("SELECT col\_name,col\_type FROM db\_postgre\_colsinfo WHERE table\_id=%(table\_id)s",  
 {"table\_id": table\_id})  
 cols\_info = cur.fetchall()  
 sql\_query = f"DELETE FROM {db\_name}" + "\_" + f"{table\_name}" \  
 f" WHERE id NOT IN (SELECT MAX(id) AS MaxRecordId FROM {db\_name}\_" + \  
 f"{table\_name} GROUP BY "  
 for i in range(len(cols\_info)):  
 sql\_query += cols\_info[i][0]  
 if i == len(cols\_info) - 1:  
 sql\_query += ");"  
 else:  
 sql\_query += ", "  
 cur.execute(sql\_query)  
 conn.commit()  
 cur.close()  
 conn.close()  
 return HttpResponse(status=200)

Далі код перписанний за допомогою схожих sql запитів