

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

ВЕБ-ЗАСТОСУНОК ДЛЯ ВІДОБРАЖЕННЯ ДАНИХ З ФАЙЛУ БАЗИ ДАНИХ

Текстова частина до курсової роботи

за спеціальністю „Комп’ютерні науки” 122

Керівник курсової роботи

с.в. Борозенний С.О.

(прізвище та ініціали)

(підпис)

“ ____ ” _____ 2023 р.

Виконав студент _____

Вакуленко В.С.

(прізвище та ініціали)

“ ____ ” _____ 2023 р.

Київ 2023

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри мультимедійних систем,

доцент, к.ф-м.н.

_____ О. П. Жежерун (підпис)

»_____» _____ 2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Вакуленко Владиславу Сергійовичу факультету інформатики 3-го курсу

ТЕМА Веб-застосунок для відображення даних з файлу бази даних

Календарний план

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу	01.11.2022	
2.	Аналіз матеріалів за темою	01.12.2022	
3.	Розробка та програмування алгоритму	01.03.2023	
4.	Написання текстової частини до курсової роботи	01.05.2023	
5.	Захист курсової роботи	22.05.2023	

Вакуленко В. С. _____

Борозенний О. С. _____

“ _____ ”

Зміст

Анотація.....	5
Вступ.....	6
1 СКБД та формати експорту БД.....	8
1.1 Дослідження ринку і виявлення основних СКБД.....	8
1.2 Дослідження форматів експорту БД у конкретних СКБД.....	10
1.3 Результати дослідження і постановка задачі для розробки парсеру.....	13
2 Парсер.....	15
2.1 Алгоритм.....	15
2.2 Проблеми.....	15
2.2.1 Типи даних.....	15
2.2.2 Дані.....	17
3 Веб-застосунок.....	19
3.1 Інструменти та технології, використані в застосунку.....	19
3.2 Огляд функціоналу.....	22
Висновки.....	27
Перелік прийнятих скорочень.....	28
Список використаних джерел.....	28

Анотація

Робота спрямована на розробку веб-застосунку, який відображає дані з бази даних. Застосунок буде корисний людям, які захочуть відобразити дані різних форматів, використовуючи один інструмент.

У результаті розроблено веб-застосунок, який дозволяє відображати дані з бази даних. Застосунок використовує такі технології, як React [2], Nodejs [6], HTML, CSS, JavaScript, і взаємодіє з базою даних за допомогою мови запитів SQL.

Ключові слова: відображення даних, бази даних

Вступ

У сучасному світі бази даних є інструментом для зберігання великих обсягів даних і використовуються в багатьох галузях. Це означає, що потреба в перегляді даних може викинути і в людини, яка не знається на роботі з базами даних, не має інструментів для цього або просто не вміє ними керувати.

Наразі існують такі інструменти, як phpMyAdmin [17]. Вони дозволяють імпортувати файл бази даних і побачити дані. Проте такі інструменти потрібно встановлювати локально, а відображення даних має табличний вигляд. Виходячи з поставленої проблеми, ці інструменти не зможуть вирішити її.

Зручнішим рішенням міг би бути веб-застосунок, який би дозволяв відобразити дані, наприклад, за аналогією до сіток інтернет магазинів. Таких рішень наразі не існує.

Метою цієї роботи є розробка веб-застосунку, який би дозволяв відобразити дані, наприклад, за аналогією до сіток інтернет магазинів. Дослідження та розробка такого веб-застосунку дозволить вирішити поставлену проблему, а також стане основою для моєї дипломної роботи.

Новизна теми визначається потенціалом та потребою у вирішенні проблеми зручності відображення даних з бази даних.

Актуальність цієї роботи полягає в важливості для людей мати доступ до перегляду даних з різних баз даних у зручному вигляді, незалежно від досвіду цих людей у роботі з базами даних та без витрати великої кількості часу. Цей веб-застосунок може

забезпечити можливість зручного доступу до даних для різних категорій користувачів.

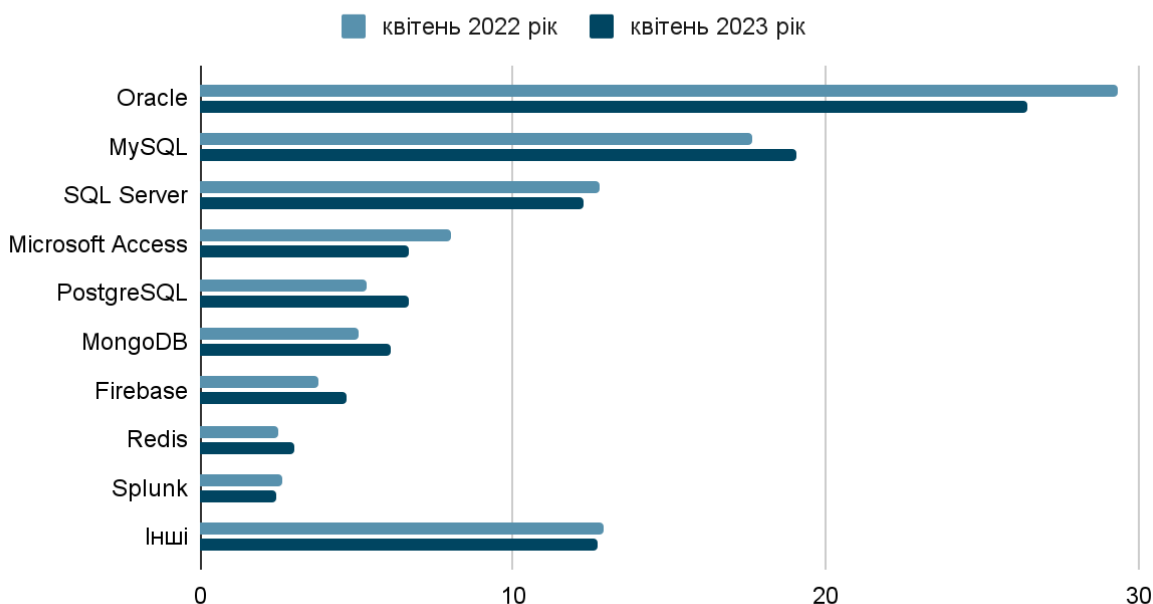
Основна частина

1 СКБД та формати експорту БД

1.1 Дослідження ринку і виявлення основних СКБД

Для визначення найпопулярніших СКБД, для яких я писатиму парсер, я обрав TOP Database index [10]. TOPDB - це показник за яким обраховують популярність СКБД та який обрахований за кількістю згадувань в пошуку Google. Це не найкраща метрика для вибору СКБД для використання в проекті, проте дуже показова для цілей цього проекту. Ці дані корисні для даного проекту, оскільки він створений саме для використання пересічними людьми, які і створюють основну частину пошукових запитів. Отже попит на користування саме цими СКБД найвищий.

TOPDB Top Database index



Гістограма 1.1 - TOPDB індекс 2022/2023 років [10]

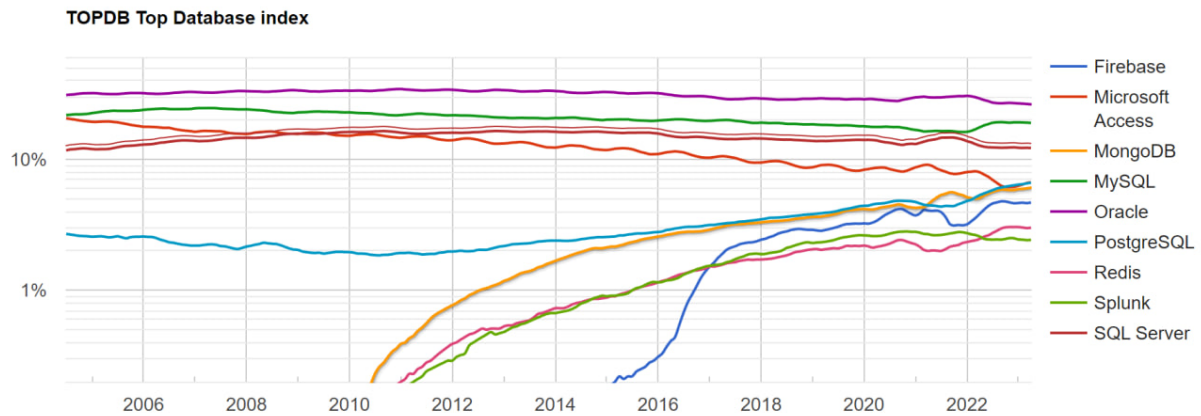


Рисунок 1.1 - динаміка TOPDB індекс за 17 років [10]

Згідно з даними на квітень 2023 року (гістограма 1.1) та даними за останні 17 років (рисунок 1.1), Oracle [11], MySQL [12] та SQL Server [13] є найпопулярнішими СКБД. Отже для них я проводитиму подальшу роботу.

1.2 Дослідження форматів експорту БД у конкретних СКБД

```

1  -----
2  -- DDL for Table CATEGORY
3  -----
4
5  CREATE TABLE "SYS"."CATEGORY"
6  ( "CATEGORY_NUMBER" NUMBER(*,0),
7  "CATEGORY_NAME" VARCHAR2(50 BYTE)
8  ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
9  NOCOMPRESS LOGGING
10 STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
11 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
12 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
13 TABLESPACE "SYSTEM" ;
14 REM INSERTING into SYS.CATEGORY
15 SET DEFINE OFF;
16 Insert into SYS.CATEGORY (CATEGORY_NUMBER,CATEGORY_NAME) values (2,'Фрукти');
17 Insert into SYS.CATEGORY (CATEGORY_NUMBER,CATEGORY_NAME) values (1,'Овочі');
18 Insert into SYS.CATEGORY (CATEGORY_NUMBER,CATEGORY_NAME) values (5,'Бакалія');
19 Insert into SYS.CATEGORY (CATEGORY_NUMBER,CATEGORY_NAME) values (8,'Хліб');
20 -----
21 -- DDL for Index CATEGORY_PK
22 -----
23
24 CREATE UNIQUE INDEX "SYS"."CATEGORY_PK" ON "SYS"."CATEGORY" ("CATEGORY_NUMBER")
25 PCTFREE 10 INITRANS 2 MAXTRANS 255
26 STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
27 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
28 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
29 TABLESPACE "SYSTEM" ;
30 -----
31 -- Constraints for Table CATEGORY
32 -----
33
34 ALTER TABLE "SYS"."CATEGORY" MODIFY ("CATEGORY_NUMBER" NOT NULL ENABLE);
35 ALTER TABLE "SYS"."CATEGORY" MODIFY ("CATEGORY_NAME" NOT NULL ENABLE);
36 ALTER TABLE "SYS"."CATEGORY" ADD CONSTRAINT "CATEGORY_PK" PRIMARY KEY ("CATEGORY_NUMBER")
37 USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
38 STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
39 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
40 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
41 TABLESPACE "SYSTEM" ENABLE;

```

Рисунок 1.2 - Oracle СКБД SQL формат експорту

```
1  --
2  -- Table structure for table `category`
3  --
4
5  CREATE TABLE `category` (
6    `category_number` int(11) NOT NULL,
7    `category_name` varchar(50) NOT NULL
8  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
9
10 --
11 -- Dumping data for table `category`
12 --
13
14 INSERT INTO `category` (`category_number`, `category_name`) VALUES
15 (1, 'Овочі'),
16 (2, 'Фрукти'),
17 (5, 'Бакалія'),
18 (8, 'Хліб');
19
20 --
21 -- Indexes for table `category`
22 --
23 ALTER TABLE `category`
24   ADD PRIMARY KEY (`category_number`);
25
26 --
27 -- AUTO_INCREMENT for table `category`
28 --
29 ALTER TABLE `category`
30   MODIFY `category_number` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=12;
```

Рисунок 1.3 - MySQL СКБД SQL формат експорту

```

1  <?xml version="1.0" encoding="utf-8"?>
2
3  <pma_xml_export version="1.0" xmlns:pma="https://www.phpmyadmin.net/some_doc_url/">
4    <pma:structure_schemas>
5      <pma:database name="test-ais" collation="utf8mb4_general_ci" charset="utf8mb4">
6        <pma:table name="category">
7          CREATE TABLE `category` (
8            `category_number` int(11) NOT NULL AUTO_INCREMENT,
9            `category_name` varchar(50) NOT NULL,
10           PRIMARY KEY (`category_number`)
11          ) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8mb4;
12        </pma:table>
13      </pma:database>
14    </pma:structure_schemas>
15
16    <database name="test-ais">
17      <table name="category">
18        <column name="category_number">1</column>
19        <column name="category_name">Овочі</column>
20      </table>
21      <table name="category">
22        <column name="category_number">2</column>
23        <column name="category_name">Фрукти</column>
24      </table>
25      <table name="category">
26        <column name="category_number">5</column>
27        <column name="category_name">Бакалія</column>
28      </table>
29      <table name="category">
30        <column name="category_number">8</column>
31        <column name="category_name">Хліб</column>
32      </table>
33    </database>
34  </pma_xml_export>

```

Рисунок 1.4 - MySQL СКБД XML формат експорту

```

1 {
2   "outputs": [],
3   "execution_count": 0,
4   "cell_type": "code",
5   "source": [
6     "/****** Object: Table [dbo].[category]   Script Date: 4/23/2023 6:14:02 *****/\r\nSET ANSI_NULLS ON\r\n",
7     "GO\r\n",
8     "SET QUOTED_IDENTIFIER ON\r\n",
9     "GO\r\n",
10    "CREATE TABLE [dbo].[category](\r\n\t[category_number] [int] NOT NULL,\r\n\t[category_name] [nvarchar](50) NOT NULL\r\n) ON [PRIMARY]\r\nGO\r\n",
11    ],
12    "metadata": {
13      "urn": "Server[@Name='DESKTOP-GKGL62']/Database[@Name='test-ais']/Table[@Name='category' and @Schema='dbo']",
14      "object_type": "Table"
15    }
16  },
17  {
18    "outputs": [],
19    "execution_count": 0,
20    "cell_type": "code",
21    "source": [
22      "INSERT [dbo].[category] ([category_number], [category_name]) VALUES (1, N'Овочі')\r\n",
23      "INSERT [dbo].[category] ([category_number], [category_name]) VALUES (2, N'Фрукти')\r\n",
24      "INSERT [dbo].[category] ([category_number], [category_name]) VALUES (5, N'Бакалія')\r\n",
25      "INSERT [dbo].[category] ([category_number], [category_name]) VALUES (8, N'Хліб')\r\n",
26      "GO\r\n",
27    ],
28    "metadata": {
29      "urn": "Server[@Name='DESKTOP-GKGL62']/Database[@Name='test-ais']/Table[@Name='category' and @Schema='dbo']",
30      "object_type": "Table"
31    }
32  },

```

Рисунок 1.5 - SQL Server СКБД Ipython формат експорту

```

1 CREATE TABLE [dbo].[category](
2     [category_number] [int] NOT NULL,
3     [category_name] [nvarchar](50) NOT NULL
4 ) ON [PRIMARY]
5 GO
6 INSERT [dbo].[category] ([category_number], [category_name]) VALUES (1, N'Овочі')
7 INSERT [dbo].[category] ([category_number], [category_name]) VALUES (2, N'Фрукти')
8 INSERT [dbo].[category] ([category_number], [category_name]) VALUES (5, N'Бакалія')
9 INSERT [dbo].[category] ([category_number], [category_name]) VALUES (8, N'Хліб')

```

Рисунок 1.6 - SQL Server СКБД SQL формат експорту

1.3 Результати дослідження і постановка задачі для розробки парсеру

Отже, виходячи зі зробленого дослідження, основний формат експорту баз даних з обраних СКБД - це SQL. А також для деяких СКБД наявні інші зручні формати експорту, як-от: Ipython для SQL Server та XML для MySQL. Хочу зазначити, що в обраних СКБД є й

інші формати експорту баз даних, але вони менш зручні або не мають повної інформації про базу даних.



Рисунок 1.7 - Структура вихідного документу

2 Парсер

2.1 Алгоритм

- a) отримати файл;
- b) визначити розширення файлу;
- c) прочитати файл;
- d) запустити парсер для відповідного розширення;
- e) виділити структуру таблиць;
- f) визначити зі структури таблиці назви стовпців, типи даних та обмеження на NULL;
- g) змінити тип даних, якщо наявного немає в MySQL;
- h) виділити рядки модифікування таблиць;
- i) визначити ключі і первинні ключі;
- j) виділити рядки додавання даних;
- k) змінити типи даних відповідно до схеми;
- l) виділити рядки додавання обмежень;
- m) зберегти всі отримані дані, як JSON об'єкт;
- n) створити MySQL файл для запиту до БД.

2.2 Проблеми

2.2.1 Типи даних

Порівнюючи типи даних в MySQL з типами даних у Oracle та SQL Server, можна умовно виділити 3 види:

- a) типи даних власне MySQL чи типи даних інших СКБД, що мають повних відповідників у MySQL за синтаксисом та суттю;
- b) типи даних інших СКБД, які мають відповідників за суттю, але відрізняються синтаксисом;

- с) типи даних інших СКБД, які не мають відповідників у MySQL.

Отже опишемо алгоритми для визначення правильного типу даних у MySQL:

- а) виділимо тип даних, який зазначений в файлі (для цього використаємо завчасно створені об'єкти для кожної СКБД, де ключами є назви типів, а значеннями - регулярні вирази для виділення цих типів з рядку);
- б) якщо цей тип наявний у MySQL, то:
 - 1) перевіримо чи коректні параметри цього типу (наприклад в Oracle параметри довжини деяких типів даних мають приписку "BYTE", що неприпустимо для MySQL) і виправимо синтаксис за потреби;
 - 2) повернемо правильний тип даних та значення (назва типу даних разом з усіма параметрами);
- с) якщо цей тип не наявний у MySQL, але має свого відповідника, то:
 - 1) дізнаємося тип даних у MySQL, який відповідає даному типу даних (для цього використаємо завчасно створені об'єкти для кожної СКБД, де ключами є назви типів у цій СКБД, а значеннями - назви типів у MySQL);
 - 2) дістанемо параметри з даного типу даних і провалідуємо їх відповідно до синтаксису MySQL;
 - 3) повернемо правильний тип даних (назва у MySQL) та значення (назва у MySQL та параметри);
- д) якщо цей тип не має відповідника у MySQL, то:
 - 1) повернемо строковий тип даних.

2.2.2 Дані

З даними є дві основні проблеми:

- а) їх правильне розбиття;
- б) відповідність типу цих даних тому типу даних, який вказаний в схемі.

Проблема з розбиттям даних полягає в тому, що ми читаємо файл будь-якого розширення як строку. Отже і дані, які в SQL описуються в круглих дужках через кому, ми отримуємо також як строку. Це призводить до того, що нам треба якось правильно розбити цю строку. Логічно було б зробити це за комою (,). Проте в такому випадку ми ризикуємо отримати неправильне розбиття, якщо якісь строкові дані міститимуть кому. Тому, враховуючи це, пропоную розбивати дані тільки за тими комами (,), які не стоять всередині лапок.

Отже опишемо алгоритм для правильного розбиття:

1. Розіб'ємо дані за комою (,)
2. Якщо довжина отриманого масиву співпадає з кількістю стовпців в таблиці, то:
 - 2.1. Повернемо отриманий масив значень
3. Якщо довжина отриманого масиву не співпадає з кількістю стовпців в таблиці, то:
 - 3.1. Створемо об'єкт для відслідковування лапок, який міститиме наступні дані:
 - 3.1.1. Масив попередніх лапок
 - 3.1.2. Символ поточного типу лапок
 - 3.1.3. Числовий лічильник
 - 3.2. Розіб'ємо рядок з даними на масив символи і проітеруємося за ним

3.3. Для кожного символу перевіримо:

3.3.1. Якщо поточний символ - це лапки, то:

3.3.1.1. Якщо поточний символ дорівнює символу поточного типу лапок з об'єкта, то:

3.3.1.1.1. В об'єкті поточному типу лапок присвоємо останнє значення з масиву попередніх

3.3.1.1.2. Лічильник зменшимо на 1

3.3.1.2. Якщо поточний символ дорівнює останньому символу типу лапок з масиву попередніх, то:

3.3.1.2.1. В об'єкті поточному типу лапок присвоємо передостаннє значення з масиву попередніх

3.3.1.2.2. Лічильник зменшимо на 2

3.3.1.3. В іншому випадку:

3.3.1.3.1. Додамо в кінець масиву попередніх значень поточне значення з об'єкту

3.3.1.3.2. В поточне значення в об'єкті запишемо поточний символ

3.3.1.3.3. Лічильник збільшимо на 1

3.3.2. Якщо поточний символ - це кома і значення лічильнику в об'єкті дорівнює 0, то:

3.3.2.1. Додамо індекс поточного символу в масиві в масив індексів

3.4. Проітеруємося за отриманим масивом індексів і розіб'ємо рядок даних правильно

3.5. Повернемо правильний масив значень

На цьому етапі програма правильно розбила дані, та проблема з типами залишилася. Вона впливає все з того ж читання файлу, де всі дані, навіть якщо мали числовий, булевий, або інші типи, прочиталися як строка.

Отже опишемо алгоритм для конвертації даних у правильний тип:

- а) отримаємо комірку даних та схему, де вказаний тип даних для цієї комірки;
- б) для кожного виду типу даних змінимо дані (для цього використаємо завчасно створений об'єкт, де ключами є види типів даних в MySQL (numeric, string, date and time, etc.), а значеннями - масиви типів даних);
- с) повернемо дані в правильному типі.

3 Веб-застосунок

3.1 Інструменти та технології, використані в застосунку

Оскільки, виходячи з поставленої проблеми, важливими критеріями якості роботи цього інструменту є швидкий доступ до перегляду даних бази даних без потреби будь-якого досвіду у роботі з базами даних, то веб-застосунок є найкращим варіантом за такими критеріями:

- а) легкий та швидкий доступ з будь-якого пристрою, що не вимагає завантаження ПЗ;
- б) зрозуміле відображення даних у звичному форматі.

Для того, аби до застосунку дійсно можна було отримати доступ з будь-якого пристрою, його потрібно розгурнути на хостингу. Для цього я обрав Heroku [18]. Цей хмарний хостинговий сервер має багато переваг в контексті роботи над даним застосунком:

- a) зручне та просте розгортання, а також швидке оновлення та зрозумілі інструменти відладки;
- b) наявність безкоштовних послуг, яких вистачає для розгортання і роботи над додатком в режимі розробки;
- c) можливість легкого масштабування;
- d) легка інтеграція з іншими сервісами, такими як бази даних.

А також це просто один з найпопулярніших хмарних веб хостингів серед проектів написаних на Node.js.

Розгортання сайту відбувається всього в декілька кроків:

- a) написання сайту за схемою Heroku;
- b) викладення сайту на GitHub [18] репозиторій;
- c) створення акаунту на Heroku;
- d) під'єднання GitHub акаунту до акаунту Heroku;
- e) створення проекту в Heroku, який буде прив'язаний до GitHub репозиторія;
- f) перший deploy (подальші будуть відбуватися автоматично з оновленням GitHub репозиторію).

HTML, CSS, JS.

React [2] - це бібліотека для розробки веб інтерфейсів.

Основними перевагами React є:

- a) компонентний підхід до розробки, що дозволяє перевикористовувати компоненти і тим самим зменшити кількість коду;
- b) Virtual DOM, що являє собою копію реального DOM дерева. Замість змін в DOM, React накоплює зміни у Virtual DOM, потім порівнює його з DOM та здійснює перемальовку виключно тих блоків, де відбулася зміна. Це дозволяє уникнути важких операцій звернення до DOM, а також перемальовки великої кількості блоків, де не відбулися зміни;

с) наявність великої кількості бібліотек (наприклад Redux та React Router), які легко під'єднуються за допомогою npm.

Redux [4] - менеджер станів у React. В React є таке поняття, як стани (state). Стани бувають локальні, тобто ті, які використовуються в межах однієї компоненти. Передача локальних станів в дочірні компоненти вважається невдалою практикою, тож для цього використовують менеджери станів. Для React існує два менеджери станів: Redux та MobX. Основними перевагами Redux є:

- а) простота та прозорість в розробці та тестуванні;
- б) наявність екосистеми (наприклад Redux DevTools).

React Router [5] - це інструмент для рендерингу на стороні клієнта. Такий підхід дозволяє при переході за посиланням отримувати з серверу тільки дані, що в свою чергу не викликає оновлення сторінки, менше навантажує сервер та пришвидшує роботу застосунку.

Node.js [6] - це середовище виконання JS, яке використовується в даному проекті для створення серверу за допомогою Express, підключення до бази даних за допомогою MySQL та налаштування маршрутизації.

Express [7] - веб фреймворк для Node.js.

MySQL [12] - реляційна система управління базами даних.

3. 2. Огляд функціоналу

Авторизація здійснюється з використанням Google акаунту.

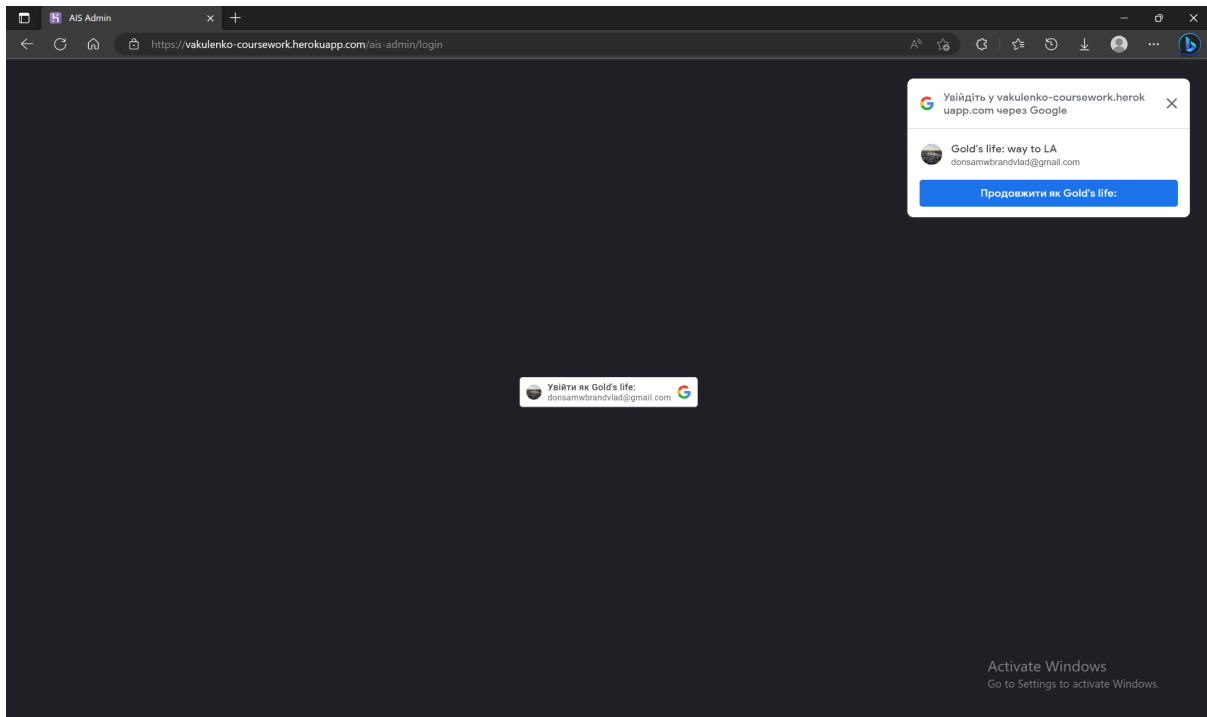


Рисунок 3.1 - Авторизація з використанням Google account

У випадку, якщо це був перший вхід користувача, то в нього буде лише можливість імпортувати базу даних.

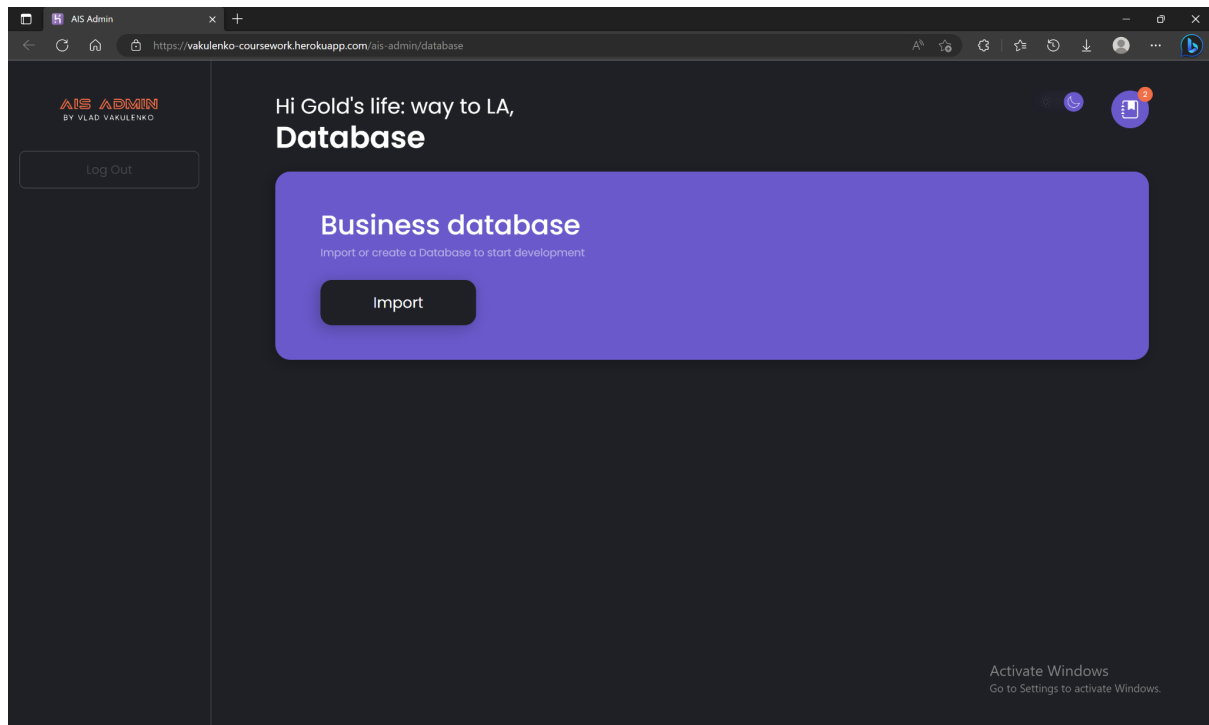


Рисунок 3.2 - Імпорт файлу бази даних

Після імпорту бази даних, користувач побачить на екрані структуру бази даних, інформацію щодо типів даних та зв'язки між сутностями. Також в меню з'являються додаткові пункти для подальшої роботи з базою даних

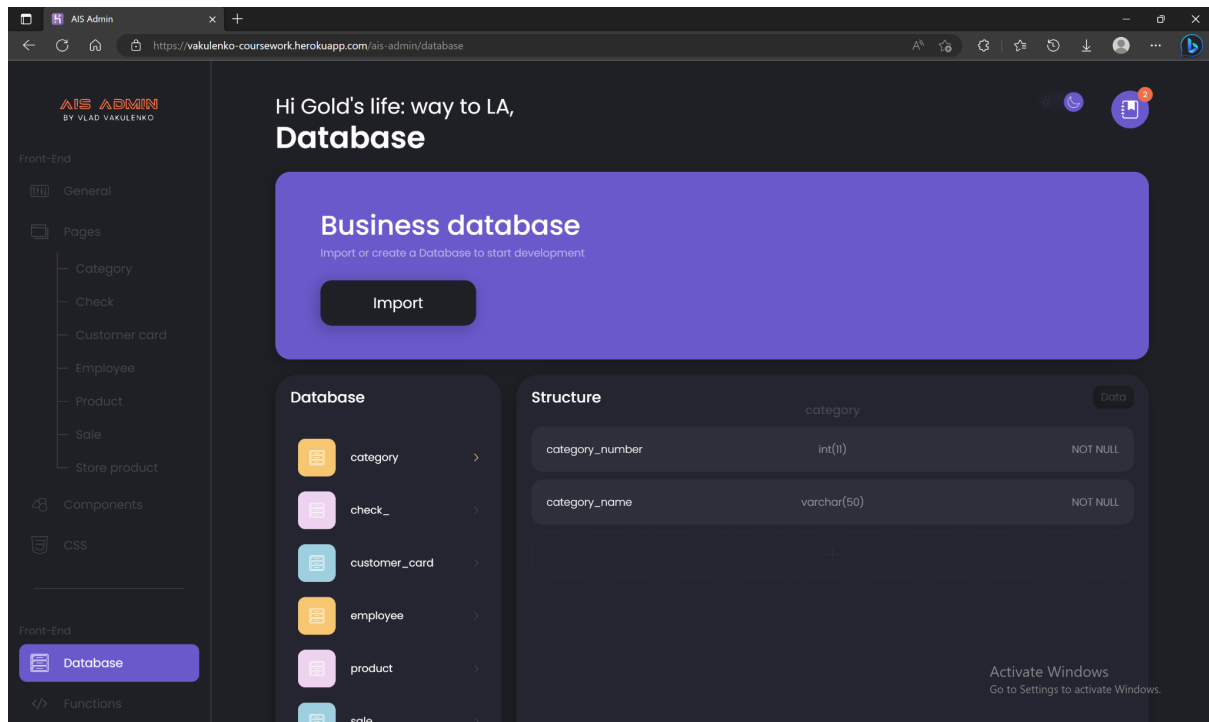


Рисунок 3.3 - Перегляд структури бази даних

Для перегляду даних користувач має перейти на головну сторінку і натиснути “Preview”

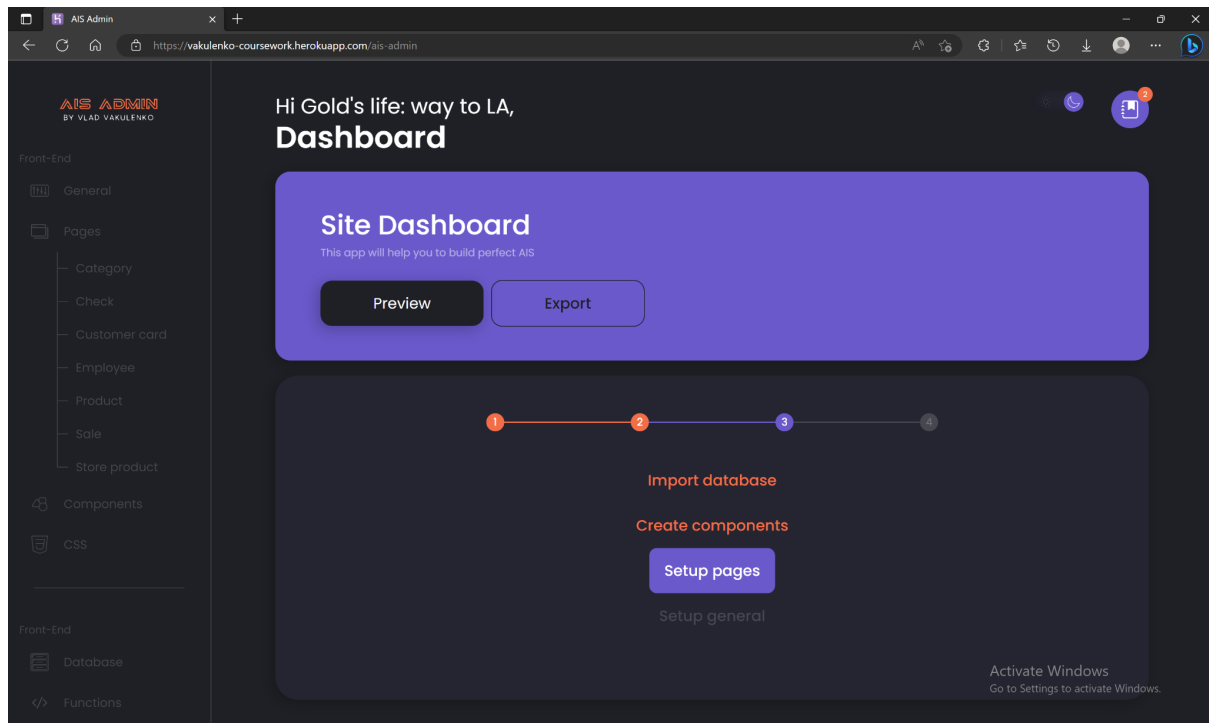


Рисунок 3.4 - Головна сторінка адмін панелі

В “Preview” користувач може переміщуватися між таблицями, використовуючи пункти меню. На кожній сторінці користувач може переглядати всі дані.

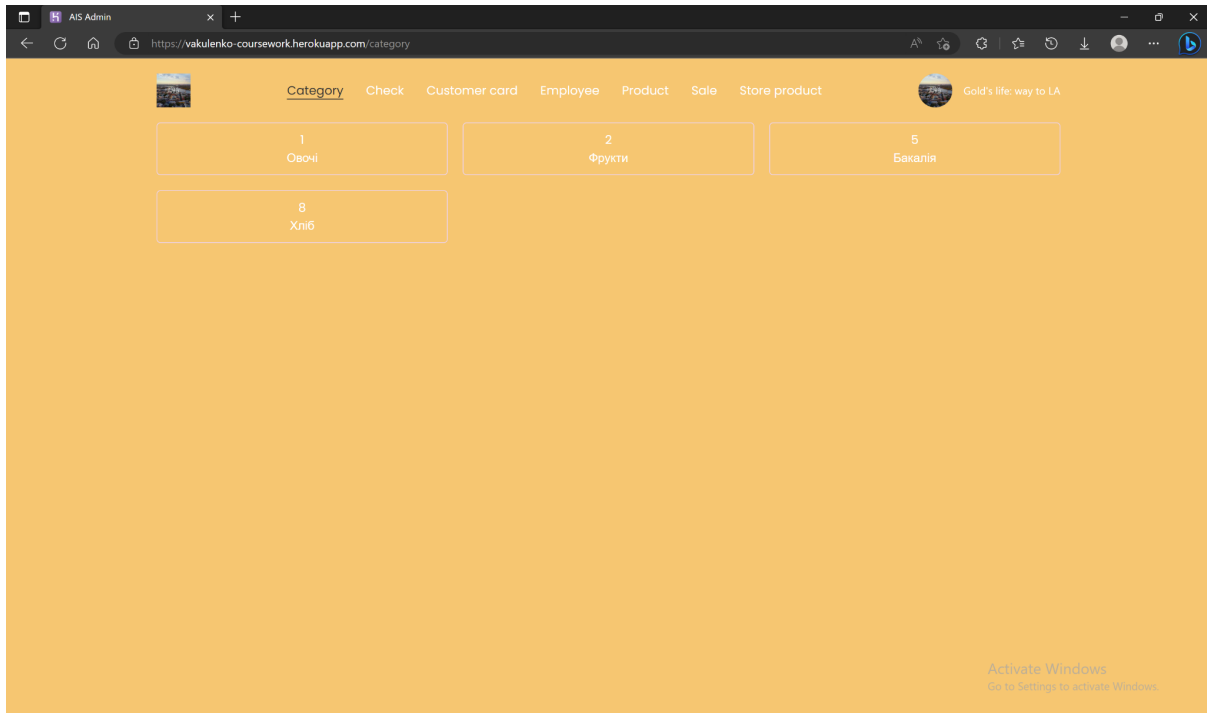


Рисунок 3.5 - Перегляд даних

Висновки

В процесі виконання цієї роботи був розроблений парсер, який дозволяє читати файли таких баз даних, як Oracle, SQL Server та MySQL у форматах SQL, Ipython та XML. Також для зручного використання цього парсеру був розроблений веб інтерфейс.

Цей застосунок буде корисний для людей, які не мають досвіду використання СКБД, не хочуть встановлювати локальні сервери для перегляду даних і хочуть відобразити дані в більш звичному вигляді.

Покращення проекту можливе в двох напрямках. Перший - це розширення існуючого функціоналу за рахунок можливостей додавання фільтрів, а також виконання операцій CRUD над даними. Другий - це вдосконалення проекту до застосунку, який дозволить без коду змінювати візуальну і функціональну складові проекту, тобто створювати власні компоненти для відображення даних, власний функціонал для кнопок, тощо.

Перелік прийнятих скорочень

БД - база даних

СКБД - система керування базою даних

Список використаних джерел

- [1] [detect-file-encoding-and-language - npm \(npmjs.com\)](https://www.npmjs.com/package/detect-file-encoding-and-language)
- [2] <https://react.dev/>
- [3] <https://www.npmjs.com/>
- [4] <https://redux.js.org/>
- [5] <https://reactrouter.com/en/main>
- [6] <https://nodejs.org/en>
- [7] <https://expressjs.com/>
- [8] <https://redux-toolkit.js.org/>
- [9] [The Most Popular Databases – 2006/2022 - \(statisticsanddata.org\)](https://statisticsanddata.org/)
- [10] [TOPDB Top Database index \(pypl.github.io\)](https://pypl.github.io/top/)
- [11] [Database | Oracle](https://www.oracle.com/database/)
- [12] [MySQL](https://www.mysql.com/)
- [13] [Microsoft Data Platform | Microsoft](https://www.microsoft.com/data/)
- [14] [Database Software and Applications | Microsoft Access](https://www.microsoft.com/access/)
- [15] <https://www.npmjs.com/package/fs>
- [16] <https://www.npmjs.com/package/fast-xml-parser>
- [17] [phpMyAdmin](https://www.phpmyadmin.net/)
- [18] [Platform as a Service | Heroku](https://www.heroku.com/)

[19] [GitHub: Let's build from here · GitHub](#)