

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мережних технологій факультету інформатики



Розробка веб-застосунку для аналізу результатів ЗНО
Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення»

Керівник курсової роботи

Кандидат фізико-математичних наук,

Старший викладач

Гречко А. В.

(підпис)

“ ____ ” _____ 2020 р.

Виконав студент ІПЗ-3:

Липка Є. Є.

“ ____ ” _____ 2020 р.

Зміст

Вступ.....	2
Актуальність, наукове та практичне значення обраної теми.....	2
Розділ 1. Аналіз предметної області. Постановка завдання курсової роботи.....	2
Розділ 2. Опис реалізації програмного продукту.....	4
Аналіз технічного завдання	4
Обґрунтування алгоритму й структури програми	7
Обґрунтування вибору засобів розробки	9
Опис розробки програми.....	10
Створення об'єктів і розробка головної програми	12
Опис файлів даних та інтерфейсу програми.....	16
Висновки.....	20
Додаток А	21

Вступ

Актуальність, наукове та практичне значення обраної теми

Зовнішнє незалежне оцінювання проводиться щороку вже понад десятиріччя після відповідної постанови кабінету міністрів України від 31 грудня 2005 року. Детальний аналіз результатів кожного року є надзвичайно важливим як для навчальних закладів, чий престиж залежить від показників учнів, які з нього випускаються, так і для батьків, які збираються віддавати своїх дітей до школи, і мусять визначитися, яку саме обрати.

Метою цієї роботи є створення застосунку, який дозволить користувачам швидко і легко дізнаватися показники минулих років, а також отримувати рекомендації щодо найкращих навчальних закладів, які вони можуть обрати для своїх дітей, на основі середніх показників випускників.

Для розробки було використано технології HTML, CSS, мову JavaScript із програмною платформою Node.js і фреймворк Express.js. Для зберігання даних було використано MongoDB – NoSQL систему управління базами даних, яка дозволяє зберігати документи подібні на формат JSON без необхідності створювати таблиці вручну.

Розділ 1. Аналіз предметної області. Постановка завдання курсової роботи

Аналіз сучасного стану питання та обґрунтування теми

На даний момент, після кожного проведеного зовнішнього незалежного оцінювання, результати публікуються на сайті українського центру оцінювання якості освіти (УЦОЯО) в загальному доступі. Користувачі, які бажають дізнатися результати, можуть вільно це зробити на офіційному сайті УЦОЯО, однак проблема полягає в тому, що для пересічного користувача ці дані складно аналізувати в тому вигляді, в якому вони подані, а також, що більшість

із них студенту або батькам, які хочуть віддати свою дитину до школи, не надає необхідної чи корисної інформації.

Так, серед даних надаються звіти з різних опитувань учасників зовнішнього незалежного оцінювання, які не допомагають визначитися який же навчальний заклад обрати, а також загальні звіти про проведення ЗНО кожного року, кожен із яких, хоч і містить таблиці з усіма можливими даними, має загальний обсяг у понад 500 сторінок, що робить його аналіз та пошук релевантної інформації в ньому для більшості пересічних користувачів складним і тривалим. Також офіційний сайт УЦОЯО надає можливість дізнатися показники випускників в розрізі окремого предмету і окремої області або населеного пункту, однак надає для цього три різні веб-сторінки, кожна з яких дозволяє отримати корисну інформацію, але жодна не надає усієї інформації і не завжди у зручному вигляді. Так, батьки, які проживають в Луцьку і планують наступного року відправляти свою дитину до школи, яка дає найкращі знання з математики, мусять переглянути таблицю з сорока навчальними закладами, про які надається лише загальна кількість учасників ЗНО цього року і відсоток тих, хто склав даний предмет на оцінку в одному із п'яти діапазонів у двадцять балів. Таким чином, батькам необхідно буде виключити навчальні заклади із низькою кількістю учасників, адже результат у 50% учасників, які отримали оцінку від 180 до 200 балів при загальній кількості учасників у дві особи не можна вважати релевантним. Також, необхідно буде переглянути показники усіх інших навчальних закладів та самостійно порівняти їх між собою, а оскільки надана таблиця відсортована в алфавітному порядку за назвою закладу і не дозволяє будь-якого іншого сортування, такий аналіз знову виявиться складним та тривалим.

Серед інших ресурсів не було знайдено жодного, який би надавав інші можливості отримати дані в розрізі певного населеного пункту чи навчального закладу. Більшість із них або надають загальну статистику по всій Україні, або демонструють таблиці, вже доступні на офіційному сайті УЦОЯО.

Отже, після проведеного аналізу доступних ресурсів, було виявлено дві основні проблеми:

- 1) Немає ресурсу, який би надавав можливість отримати усю релевантну інформацію з однієї сторінки
- 2) Немає ресурсу, який би дозволяв батькам та учням швидко і легко визначити навчальні заклади їхнього населеного пункту, випускники яких показують найкращі результати з обраного предмету

Таким чином, було поставлено задачу створити Single Page Application (SPA), що надаватиме як загальну статистику, так і інформацію з окремих шкіл і матиме наступні можливості:

- 1) Дізнатися загальну кількість учасників по всій Україні або в конкретній області за певними параметрами: рік народження, стать, тип навчального закладу і т.д.
- 2) Дізнатися загальну статистику учасників з обраного предмету на обраній території (область, район чи населений пункт) або з обраного навчального закладу
- 3) Дізнатися і порівняти середні результати випускників із певного предмету обраної території
- 4) Отримати рекомендації щодо найкращих навчальних закладів з певного предмету в обраному населеному пункті

Розділ 2. Опис реалізації програмного продукту

Аналіз технічного завдання

Вхідні дані взято із порталу відкритих даних офіційного сайту УЦОЯО, який надає відкритий доступ до csv файлів за останні чотири роки із даними про кожного учасника та предмети, ЗНО з яких він здавав, із оцінкою за предмет (не вказуючи при цьому імені учасника). Ці дані необхідно

проаналізувати та заповнити ними базу даних для швидкого і зручного до них доступу.

Оскільки у програми має бути чотири основні функції, розглянемо кожну із них окремо, зазначивши спершу, що користувач може задати глобальний параметр Рік проведення ЗНО, результати саме з якого будуть аналізуватися в усіх наступних функціях, якщо не вказано протилежного:

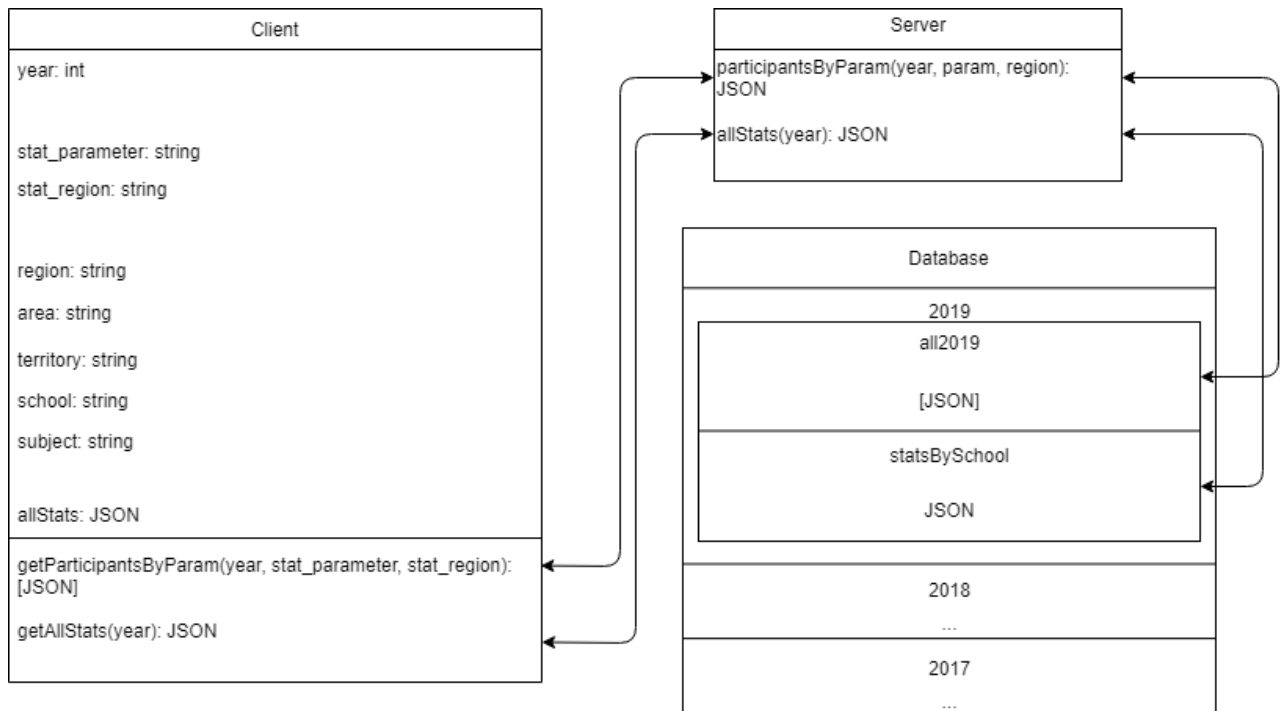
1. Користувач може отримати дані про кількість учасників по всій Україні або в обраній області за наступними параметрами: рік народження, стать, тип учасника (випускник минулих років, випускник закладу вищої освіти і т.д.), тип навчального закладу. На виході користувач мусить отримати масив об'єктів із двома властивостями: значення параметру і кількість учасників. Наприклад, для року народження [{p:2001, n:173718}, {p:2000, n:106454}, ...]. Цей масив буде переведено і надано користувачеві у вигляді гістограми. Користувач може задати три параметри: конкретну область (або всі області разом), а також один із параметрів перерахованих вище, дані за яким користувач бажає отримати дані.
2. Користувач може дізнатися кількість учасників, які не з'явилися, не подолали поріг, роботи яких було анульовано та зараховано з певного предмету на обраній території або з обраного навчального закладу, а також загальну кількість учасників та середню оцінку. На виході користувач має отримати один JSON-об'єкт із шістьма перерахованими вище властивостями. Цей об'єкт буде перетворено та надано користувачеві у вигляді гістограми. Користувач може задати два параметри: предмет для аналізу та територію, причому територія може обмежуватися як цілою областю, так і районом, населеним пунктом, або лише одним навчальним закладом.
3. Користувач може дізнатися і порівняти середні оцінки з певного предмету об'єктів обраної території. Наприклад, якщо користувач обирає Вінницьку область, то він отримає середню оцінку кожного району цієї

області, якщо користувач обирає населений пункт, то він отримає середню оцінку кожного навчального закладу цього населеного пункту. На виході користувач має отримати масив об'єктів із двома властивостями: назва адміністративної одиниці або школи та середня оцінка. Цей масив буде перетворено та надано користувачеві у вигляді гістограми, відсортованої за середньою оцінкою. Користувач може задати два параметри: предмет для аналізу та адміністративну одиницю, яка може обмежуватися як цілою областю, так і районом або населеним пунктом.

4. Користувач може отримати список найкращих або усіх навчальних закладів обраного населеного пункту із такими даними як назва навчального закладу, кількість учасників та середній бал з певного предмету. Різниця полягає в тому, що для найкращих навчальних закладів враховуються результати усіх доступних попередніх років, а не лише одного обраного, а також враховуються тільки ті навчальні заклади, кількість учасників з яких перевищує певний заданий поріг задля уникнення нерелевантних результатів. На виході користувач має отримати масив об'єктів із трьома властивостями: назва навчального закладу, загальна кількість учасників та середня оцінка. Цей масив буде перетворено та надано користувачеві у вигляді таблиці, відсортованої за середньою оцінкою у випадку найкращих шкіл або за кількістю учасників у випадку усіх шкіл. Користувач може задати два параметри: предмет для аналізу та населений пункт.

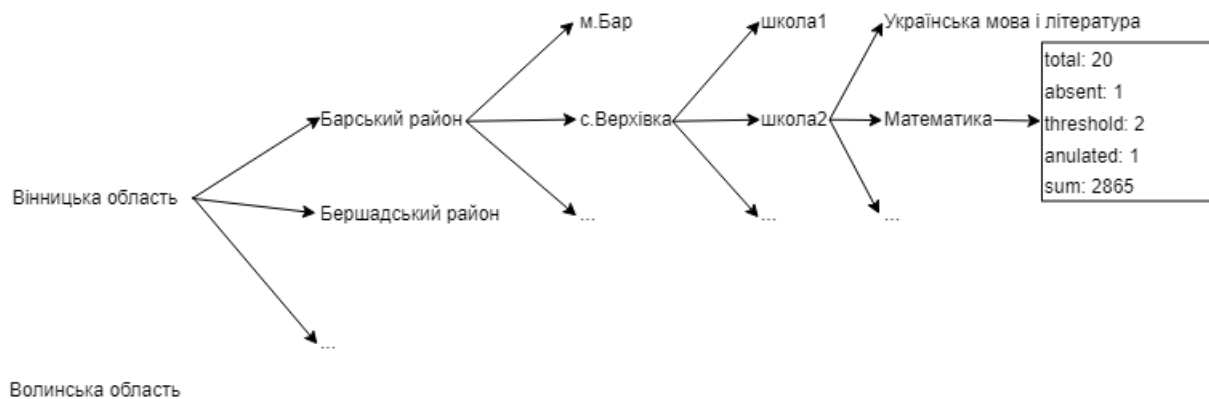
Програма буде SPA веб-застосунком, усі параметри будуть задаватися за допомогою HTML елементів таких як `input` та `select`. Взаємодія між клієнтом та сервером буде імплементована за допомогою фреймворку `Express`, усі дані зберігатимуться у базі даних `MongoDB`.

Обґрунтування алгоритму й структури програми



Малюнок 1 Діаграма роботи програми

В базі даних існує три колекції з однаковими структурами – по одній для кожного доступного року. В колекції міститиметься один великий JSON-подібний документ, який міститиме таку інформацію про кожний навчальний заклад як загальна кількість учасників, кількість тих, хто не з’явився, не подолав поріг, чия робота була анульована та сума набраних балів (для можливості порахувати середній бал). JSON-подібний документ матиме структуру дерева, де на верхівці будуть усі області. Кожна область матиме своє піддерево, у якому будуть усі населені пункти області, кожен населений пункт міститиме інформацію про всі його школи, кожна школа міститиме дані про учнів за кожним предметом, перераховані вище. Таким чином можна буде швидко діставати інформацію один або всі навчальні заклади, знаючи область, район та населений пункт. В базі даних також міститиметься колекція із усіма даними про кожного учасника ЗНО та предмети, які він здавав, разом із результатами.



Малюнок 2 Структура JSON документу в базі даних

Програма розділена на клієнтську та серверну частини. На клієнтській частині при встановленні з'єднання або при зміні обраного року викликається функція, яка відправляє запит на сервер із обраним роком як параметром. Серверна частина отримує копію описаного вище JSON-подібного документу із колекції обраного року та відправляє його у відповідь клієнтові, де цей об'єкт зберігатиметься для подальшого використання як поле allStats. Таким чином більшість обрахунків надалі буде виконуватися на стороні клієнта без потреби робити запити на сервер, що дозволить швидко і динамічно оновлювати таблиці та графіки на сторінці.

На сторінці клієнта міститиметься зокрема п'ять випадających списків: область, район, населений пункт, навчальний заклад, предмет. Коли клієнт змінюватиме значення одного із цих списків, автоматично викликатиметься функція, яка оновлюватиме таблицю статистики за обраний предмет із інформацією про кількість учнів, анульованих робіт і т.д. При натисканні на відповідні кнопки будуть викликатися функція, яка створює гістограму середніх оцінок підоб'єктів обраної адміністративної одиниці, або ж функція, яка оновлює таблицю із інформацією про найкращі, або усі школи обраної адміністративної одиниці з певного предмету. Уся необхідна інформація для

щойно описаних функцій міститиметься в об'єкті allStats, тож всі обрахунки вестимуться з боку клієнта, і не буде необхідності робити запити на сервер.

На сторінці також міститимуться іще два випадających списки, за допомогою яких клієнт зможе отримати загальну статистику про проведене зовнішнє незалежне оцінювання обраного року. Перший випадających список дозволяє обрати один із чотирьох параметрів: рік народження, стать, тип учасника, тип навчального закладу. Другий випадających список дозволяє обрати усі або одну область, за якою виводитиметься статистика. Після натискання відповідної кнопки, викликатиметься функція, яка робитиме запит на сервер, передаючи рік та описані вище два параметри. Сервер робитиме пошук у базі даних в колекції відповідного року, яка містить усі дані про учасників, підраховуючи кількість учасників з кожним унікальним параметром, наприклад, кількість учасників 2000 року народження, 1999 року народження і т.д. Після цього сервер повертатиме результат клієнтові у вигляді масиву об'єктів зі значенням параметру та кількістю учасників. Цей масив на стороні клієнта буде перетворено та відображено у вигляді гістограми.

Отже, реалізацію даного проекту можна розбити на наступні компоненти:

- 1) Парсинг csv файлів, отриманих із офіційного сайту УЦОЯО, та створення бази даних
- 2) Реалізація серверу та його взаємодії із базою даних
- 3) Реалізація фронтенду клієнта та написання функцій для створення та оновлення таблиць і графіків
- 4) Реалізація запитів та взаємодії між клієнтом та сервером

Обґрунтування вибору засобів розробки

Оскільки розроблена програма буде односторінковим веб застосунком, то за основу для розробки було вирішено використовувати стек MEAN. Це потужний, безкоштовний відкритий у вільному доступі стек програмного забезпечення, який використовується для написання веб застосунків і зокрема SPA (single-page application). Складається даний стек із чотирьох компонентів:

MongoDB, Express.js, AngularJS, Node.js. Основна перевага його полягає в тому, що усі компоненти підтримують єдину мову програмування JavaScript, а тому клієнт, сервер, взаємодія між ними та між сервером і базою даних може бути написана однією мовою. Для даної програми буде використано три компоненти з MEAN: ModgoDB, Express.js та Node.js.

Опис розробки програми

Результатом розробки буде програма, яка дозволяє клієнту робити параметричні запити та отримувати певну інформацію, яка буде відображатися у простому та зручному для розуміння форматі. Жодних інших функцій дана програма не матиме, отож сервер виступатиме винятково посередником між клієнтом і базою даних, а сама база даних буде залишатися незмінною і використовуватися лише для читання, а не для запису. Отож першим кроком у розробці є саме створення бази даних.

Для початку необхідно завантажити дані із офіційного сайту УЦОЯО, де доступні csv файли окремо для кожного року із кожним учасником ЗНО цього року та інформацією про нього та предмети, які він здавав. Кожен такий файл має обсяг близько 700Мб. Ці файли мають наступний формат – шістнадцять стовпчиків з інформацією про учасника, такою як ID, рік народження, стать, мову навчання, інформацію про навчальний заклад, орган, якому він підпорядковується і т.д.[Малюнок 18]. Із цих шістнадцяти стовпчиків зберігатиметься лише одинадцять: дата народження, стать, тип учасника, область, район і населений пункт проживання учасника та його навчального закладу, тип і назва навчального закладу. Далі міститься по десять стовпчиків на кожен предмет, із якого можна здавати ЗНО, з них на кожен предмет зберігатимуться лише три: назва предмету, статус (не з'явився, не подолав поріг, анульовано, зараховано) і оцінка за шкалою 100-200, за умови, що даний учень має інформацію про цей предмет.

У файлі DBCreator.js міститься код, який виконує читання файлів та запис інформації до бази даних. Масивом `neededFields` задаються назви стовпчиків із

перших 16, які необхідно зберігати. Кожен стовпчик для предметів у csv файлі має формат 'код предмету'+'назва стовпчика', тобто стовпчики із оцінкою для української мови та математики мають вигляд UKRBALL100 і MATHBALL100. Оскільки початок назви відрізняється в залежності від предмету, а зберігатися будуть усі предмети, то необхідно задати закінчення тих стовпчиків, які необхідно зберігати. Це міститься у масиві `neededFieldsEndsWith`.

```
//columns from opendata csv file that will be utilized
let neededFields = ["BIRTH",
                    "SEXTYPENAME",
                    "REGNAME",
                    "AREANAME",
                    "TERNAME",
                    "REGTYPENAME",
                    "EONAME",
                    "EOTYPENAME",
                    "EOREGNAME",
                    "EOAREANAME",
                    "EOTERNAME"];
//UKRTEST, HISTTEST, MATHTEST, DEUTEST..
let neededFieldsEndsWith = ["TEST",
                             "TESTSTATUS",
                             "BALL100"]
```

Малюнок 3 Назви стовпчиків csv-файлу для зберігання

Для читання файлів було використано модулі Node `fs` та `readline`. При запуску викликається функція `readData`, яка читає перший рядок файлу та розбиває його на масив рядків, кожен з яких є назвою стовпчика. Кожен із цих рядків перевіряється на те, чи це один із стовпчиків, які необхідно зберегти, заданих раніше двома масивами. Їхні назви зберігаються у масиві разом із порядковим номером стовпчика, решта ігнорується. Таким чином на виході матимемо масив із назвами та індексами 44 стовпчиків (11 з даними про учня і по 3 з даними про кожен із 11 можливих предметів), а не 126, які потрібно буде зберегти в базу даних. Після цього програма зчитує по тисячу (або будь-яке інше задане число) рядків за раз та призупиняє зчитування. Кожен із них аналогічно розбивається на масив рядків значень, створюється один об'єкт на кожен рядок, де зберігаються всі значення на індексах, збережених на попередньому кроці, при чому порожні значення або значення `NULL` ігноруються. Тоді всі тисяча об'єктів записуються до бази даних, після чого

читання файлу продовжується. MongoDB – це документно-орієнтована СКБД, вона не містить таблиць, а натомість зберігає JSON-подібні документи в колекціях. Дана колекція має назву у форматі all0000[Малюнок 19], де замість 0000 вказується рік проведення ЗНО, для кожного року створюються власні колекції.

Після цього по чергово викликаються функції, які створюють іще три колекції:

- 1) headers0000 – зберігаються назви стовпчиків, які бралися до уваги, та їхні порядкові номери[Малюнок 20]
- 2) uniqueFields0000 – зберігаються всі унікальні значення кожного зі стовпчиків, які бралися до уваги[Малюнок 21 Малюнок 22]
- 3) statsBySchool0000 – зберігається деревоподібний об'єкт зі статистикою про кожний навчальний заклад, описаний у пункті Обґрунтування алгоритму й структури програми[Малюнок 23]

Дані для першої колекції вже збережені у двох масивах після прочитання першого рядку файлу. Реалізація створення наступних двох колекцій – це лише питання перебору відповідних атрибутів об'єктів, уже доданих до колекції all0000. Загальний час на читання файлів останніх трьох років та заповнення колекцій становить менше десяти хвилин.

Тепер, після створення бази даних, наступними кроками у розробці є створення власне клієнта та сервера, а зокрема – фронтенд частини клієнта, а також написання запитів для взаємодії між клієнтом та сервером і сервером та базою даних.

Створення об'єктів і розробка головної програми

Фронтенд частина клієнта була написана за допомогою технологій HTML та CSS із використанням набору інструментів Bootstrap. Користувач взаємодіє зі сторінкою за допомогою трьох типів елементів: радіокнопки, випадаючих списків та кнопок, при натисканні або зміні значень яких викликаються

відповідні функції, за допомогою атрибуту onclick або onchange. При завантаженні сторінки викликається функція ініціалізації, яка викликає п'ять допоміжних функцій:

- 1) GetUniqueFields() – робить запит на сервер та отримує усі можливі унікальні значення кожного атрибуту об'єктів з колекції all0000 обраного року. На цьому етапі потрібні лише назви областей, якими заповнюються два відповідні випадаючі списки.
- 2) GetStatsBySchool() – робить запит на сервер та отримує описаний вище деревоподібний JSON-об'єкт із даними про кожну школу обраного року. Використовуючи ці дані далі проводитимуться обрахунки та оновлюватимуться таблиці загальної статистики за предметом і статистики окремо по кожній школі.
- 3) GetSubjects() – робить запит на сервер та отримує список із усіма предметами, які здавали цього року. Елементами із цього списку заповнюється відповідний випадаючий список.
- 4) GetAllStats() – робить запит на сервер та отримує описаний вище деревоподібний JSON-об'єкт із сумарними даними про кожну школу за усі доступні роки. Використовуючи ці дані далі проводитимуться обрахунки та оновлюватиметься таблиця із найкращими вищими навчальними закладами.
- 5) initNumPplBySelect() – заповнює відповідний випадаючий список заздалегідь визначеними параметрами, за якими можна дізнатися кількість учасників з однієї або усіх областей (рік народження, стать, тип учасника, тип навчального закладу).

Усі вибори елементів DOM, їхнє заповнення та запити на сервер реалізовані за допомогою бібліотеки jQuery.

При зміні значення радіокнопки, якою користувач задає рік, викликаються перші три із цих п'яти функцій та отримують дані для нового

обраного року, а також видаляються всі дані із таблиць та графіки, якщо такі були.

На сервері написані відповідні чотири функції проміжної обробки Express для перших чотирьох запитів. Запити до бази даних в MongoDB дуже прості – необхідно лише вказати назву бази та назву колекції, із якої необхідно отримати інформацію.

```
app.get("/api/uniqueFields", function(req, res){
  dbClient.db("ZNO").collection("uniqueFields" + req.query.year).find({}).toArray(function(err, result){
    if(err) throw(err);

    res.send(result);
  });
});
```

Малюнок 4 Запит до бази даних

На сервері також є ще одна функція проміжної обробки, яка викликається, коли клієнт хоче знайти кількість учасників з однієї або усіх областей за певним параметром. У функцію передаються параметр та область, обрані користувачем, і виконується параметричний запит до бази даних. Для цього у функцію find() необхідно передати об'єкт-фільтр, результатом запиту будуть лише ті об'єкти, які містять усі властивості з такими ж значеннями, як і переданий об'єкт-фільтр. Тож спершу знаходимо всі унікальні значення із колекції uniqueFields0000, яких може набувати обраний користувачем параметр та записуємо їх у масив.

```
dbClient.db("ZNO").collection("uniqueFields" + req.query.year).findOne({name: param}, function(err, result){
  if(err) throw(err);

  let x1 = result.data;
  let i = 0;
  let x = [];
  x1.forEach(function(item){
    if(item){
      x[i] = item;
      i++;
    }
  });
});
```

Малюнок 5 Параметричний запит до бази даних 1

Після чого створюємо об'єкт-фільтр із кожним із цих конкретних значень та знаходимо кількість учасників із таким значенням обраного параметру.

```

let y = [];
for(let i = 0; i < x.length; i++){
    let filterObj = {};
    filterObj[param] = x[i];
    if(req.query.region){
        filterObj.EOREGNAME = req.query.region;
    }
    dbClient.db("ZNO").collection("all" + req.query.year).find(filterObj).count(function(err, result2){
        if(err) throw(err);

        y[i] = result2;
    });
}

```

Малюнок 6 Параметричний запит до бази даних 2

В результаті отримуємо два масиви, де в першому на *i*-му індексі розміщується значення параметру (наприклад, “чоловіча” і “жіноча” для статі), а в другому – кількість учасників із таким значенням параметру. Ці два масиви повертаються клієнту, де за ними будується гістограма. Для цього було використано бібліотеку CanvasJS, яка дозволяє будувати гістограми різних типів, достатньо лише вказати елемент `div`, в якому гістограма буде розташовуватись, висоту, заголовки та масив значень.

```

let data = [];
for(let i = 0; i < res.x.length; i++)
    data.push({label: res.x[i], y: res.y[i]});

generalStatsChart = new CanvasJS.Chart("general_stats_hist_wrapper", {
    height: 400,
    animationEnabled: true,
    theme: "light2",
    tooltip: {
        shared: true
    },
    title: {
        text: $('#numPplBySelect').children("option:selected").text()
    },
    axisY: {
        titleFontSize: 24,
        title: "Кількість учасників"
    },
    data: [{
        name: "кількість",
        type: "column",
        dataPoints: data
    }]
});

```

Малюнок 7 побудова гістограми за допомогою CanvasJS

З клієнтського боку також є функції, що викликаються при оновленні значень одного із чотирьох випадajuчих списків, якими задаються область, район, населений пункт та навчальний заклад. На початку роботи програми

користувач може вибрати лише область, інші списки неактивні. При зміні області автоматично заповнюється наступний випадаючий список районами із цієї області, аналогічно із районами та населеними пунктами. При оновленні даних чотирьох списків, а також списку предметів, також викликається функція, яка зчитує поточні обрані предмет та адміністративні одиниці, знаходить загальну кількість учасників, кількість тих, хто не з'явився, не подолав поріг, чиї роботи було анульовано та зараховано, а також середню оцінку та записує ці дані до таблиці. Реалізовано це за допомогою бібліотеки jQuery.

Опис файлів даних та інтерфейсу програми

Оскільки база даних є логічним першим кроком даного проекту та була описана раніше, перейдемо до опису інтерфейсу програми.

Як було вказано раніше, за допомогою радіокнопки користувач може задати глобальний параметр рік, за яким вестиметься аналіз.



Малюнок 8 Радіокнопка вибору року

Далі користувач може обрати область та параметр, і вивести на екран гістограму кількості учасників ЗНО обраного року за вказаним параметром.

Загальна статистика

Дізнайтеся загальну кількість учасників із однієї або усіх областей за обраним параметром

Параметр	Тип учасника
Область	Усі
<input type="button" value="Обрахувати"/>	

Малюнок 9 Випадаючі списки загальної статистики

На сторінці містяться чотири випадаючих списки для області, району, населеного пункту та школи, причому кожен із останніх трьох недоступний, допоки не буде обрано значення попереднього, а також предмету.

Статистика за предметом

Оберіть територію та/або навчальний заклад, а також предмет для аналізу даних.

Область	Район
Усі	
Населений пункт	Навчальний заклад
Предмет	
Обрати предмет	

Всього	Не з'явився	Не подолав поріг	Анульовано	Зараховано	Середня оцінка
--------	-------------	------------------	------------	------------	----------------

Малюнок 10 Випадаючі списки статистики за предметом

При оновленні обраного значення будь-якого із них автоматично оновлюватиметься і таблиця статистики.

Під таблицею розташовані кнопки, які дозволяють побудувати або приховати графік середніх оцінок з обраного предмету на вказаній території. Якщо обрана лише область, то будуватиметься графік середніх оцінок усіх районів області, якщо вказаний район – то усіх населених пунктів району.

Побудувати графік середніх оцінок з обраного предмету на обраній території:

Графік середніх оцінок

Приховати

Малюнок 11 Кнопки для побудови графіку середніх оцінок

Нижче розташовані дві кнопки, за допомогою яких користувач може вивести статистику усіх навчальних закладів з обраного предмету, або до десяти із найкращими середніми оцінками. Кнопки недоступні, якщо не обрано предмет або населений пункт.

Дізнатися назви навчальних закладів обраного населеного пункту із найвищими середніми показниками з обраного предмету. До уваги беруться дані за останні три роки, а також враховуються лише навчальні заклади з не менш ніж п'ятьма учасниками

Дізнатися загальну кількість учасників та середні бали з усіх навчальних закладів обраного населеного пункту з обраного предмету. До уваги беруться дані обраного року

Найкращі навчальні заклади

Усі навчальні заклади

Малюнок 12 Кнопки для відображення таблиці найкращих навчальних закладів

Тестування програми і результати її виконання

Як результат виконання програми буде наведено таблиці та графіки, які можна отримати під час її роботи.

Гістограма кількості учасників за параметром стать у Вінницькій області:



Малюнок 13 Гістограма кількості учасників

Таблиця статистики обраної школи з Історії України:

Статистика за предметом

Оберіть територію та/або навчальний заклад, а також предмет для аналізу даних

Область

Район

Дніпропетровська область

Апостолівський район

Населений пункт

Навчальний заклад

м.Апостолове

Апостолівська загальноосвітня школа І-ІІІ ступенів № 1

Повна назва навчального закладу:

Апостолівська загальноосвітня школа І-ІІІ ступенів № 1

Предмет

Історія України

Всього	Не з'явився	Не подолав поріг	Анульовано	Зараховано	Середня оцінка
14	0	0	0	14	139.5

Малюнок 14 Таблиця статистики

Графік середніх оцінок учасників ЗНО з Історії України із навчальних закладів Барського району Вінницької області:

Побудувати графік середніх оцінок з обраного предмету на обраній території:

Графік середніх оцінок Приховати



Малюнок 15 Графік середніх оцінок

Таблиця навчальних закладів із найкращими середніми результатами за останні три роки з Історії України в Місті Бар Барського району Вінницької області:

Дізнатися назви навчальних закладів обраного населеного пункту із найвищими середніми показниками з обраного предмету. До уваги беруться дані за останні три роки, а також враховуються лише навчальні заклади з не менш ніж п'ятьма учасниками

Дізнатися загальну кількість учасників та середні бали з усіх навчальних закладів обраного населеного пункту з обраного предмету. До уваги беруться дані обраного року

Найкращі навчальні заклади			Усі навчальні заклади		
Всього Учасників	Середній бал	Назва навчального закладу			
10	163.56	Барський навчально-виховний комплекс Загальноосвітня школа І-ІІІ ступенів №2-гімназія Барської міської ради Вінницької області			
13	157	Заклад Навчально-виховний комплекс Загальноосвітня школа І-ІІІ ст.№2-гімназія м.Бара			
17	156.82	Барський ліцей № 2 Барської міської ради Вінницької області			
32	153.6	Барська загальноосвітня школа І-ІІІ ступенів №1 Барської міської ради Вінницької області			
32	152.77	Барська загальноосвітня школа І-ІІІ ступенів №4 Барської міської ради Вінницької області			
12	144.64	Заклад Загальноосвітня школа І-ІІІ ст №1 м.Бара			
107	140.35	Обласний гуманітарний ліцей-інтернат для обдарованих дітей при Барському гуманітарно-педагогічному коледжі ім. Михайла Грушевського			
23	140.13	Заклад Загальноосвітня школа І-ІІІ ст №3 м.Бара			
7	134.6	Заклад Загальноосвітня школа І-ІІІ ст №4 м.Бара			

Малюнок 16 Таблиця найкращих навчальних закладів

Таблиця результатів усіх навчальних закладів в обраний рік з Історії України в Місті Бар Барського району Вінницької області:

Дізнатися назви навчальних закладів обраного населеного пункту із найвищими середніми показниками з обраного предмету. До уваги беруться дані за останні три роки, а також враховуються лише навчальні заклади з не менш ніж п'ятьма учасниками

Дізнатися загальну кількість учасників та середні бали з усіх навчальних закладів обраного населеного пункту з обраного предмету. До уваги беруться дані обраного року

Найкращі навчальні заклади			Усі навчальні заклади		
Всього Учасників	Середній бал	Назва навчального закладу			
109	125.9	Барський гуманітарно-педагогічний коледж імені Михайла Грушевського			
49	112.52	Державний навчальний заклад Барський професійний будівельний ліцей			
33	143.85	Обласний гуманітарний ліцей-інтернат для обдарованих дітей при Барському гуманітарно-педагогічному коледжі ім. Михайла Грушевського			
19	162.68	Барська загальноосвітня школа І-ІІІ ступенів №4 Барської міської ради Вінницької області			
17	156.82	Барський ліцей № 2 Барської міської ради Вінницької області			
16	159.93	Барська загальноосвітня школа І-ІІІ ступенів №1 Барської міської ради Вінницької області			
5	134.5	Барська загальноосвітня школа І-ІІІ ступенів №3 Барської міської ради Вінницької області			

Малюнок 17 Таблиця усіх навчальних закладів

Висновки

В результаті виконання роботи було розроблено односторінковий веб-застосунок, який надає можливість дізнаватися статистичну інформацію про зовнішнє незалежне оцінювання останніх років. Застосунок дозволяє отримувати інформацію про загальну кількість учасників за різними параметрами в обраній області, середню успішність з обраного предмету учнів конкретного навчального закладу або усіх навчальних закладів обраної адміністративної одиниці, отримувати рекомендації щодо найкращих навчальних закладів обраного міста в розрізі конкретного предмету. Цей застосунок може використовуватися як і тими, хто бажає просто дізнатися загальну інформацію про ЗНО чи порівняти результати різних років, так і батьками, які здаватимуть своїх дітей до школи, та випускниками дев'ятих класів у пошуках коледжу, для допомоги їм із вибором навчального закладу.

Додаток А

OutID	ID
Birth	Рік народження
SexTypeName	Стать
Regname	Регіон реєстрації/проживання учасника
AreaName	Район/Місто реєстрації/проживання учасника
TerName	Населений пункт реєстрації/проживання учасника
RegTypeName	Статус учасника
TerTypeName	Тип території
ClassProfileName	Профіль навчання учасника/ОКР, який учасник здобуває
ClassLangName	Мова навчання учасника
EOName	Заклад освіти учасника
EOTypename	Тип закладу освіти
EOREGName	Регіон, де розташований заклад освіти
EOAreaName	Район/Місто, де розташований заклад освіти
EOTerName	Населений пункт, де розташований заклад освіти
EOParent	Орган, якому підпорядковується заклад освіти
Test	Назва навчального предмета
Lang	Мова складання тесту
TestStatus	Результат складання тесту
DPALevel	Рівень складності завдань ДПА
Ball100	Оцінка за шкалою 100-200
Ball12	Оцінка за ДПА
Ball	Тестовий бал
AdaptScale	Установлення адаптивного порога
0	не потребує
3	потребує (угорська мова навчання)
4	потребує (молдовська мова навчання)
7	потребує (румунська мова навчання)
PTName	Пункт проведення ЗНО
PTRegName	Регіон, де розташований пункт проведення ЗНО
PTAreaName	Район\Місто, де розташований пункт проведення ЗНО
PTTerName	Населений пункт, де розташований пункт проведення ЗНО

Малюнок 18 Таблиця зі структурою csv файлу

```
{
  "_id" : ObjectId("5eb69d835d3fbc4e542cdc6e"),
  "BIRTH" : "2001",
  "SEXTYPENAME" : "жіноча",
  "REGNAME" : "Тернопільська область",
  "AREANAME" : "Тернопільська область",
  "TERNAME" : "м.Тернопіль",
  "REGTYPENAME" : "Студент закладу вищої освіти",
  "EONAME" : "Приватний вищий навчальний заклад Медичний коледж",
  "EOTYPENAME" : "заклад вищої освіти",
  "EOREGNAME" : "Тернопільська область",
  "EOAREANAME" : "м.Тернопіль",
  "EOTERNAME" : "м.Тернопіль",
  "UKRTEST" : "Українська мова і література",
  "UKRTESTSTATUS" : "Зараховано",
  "UKRBALL100" : "166,0",
  "HISTTEST" : "Історія України",
  "HISTTESTSTATUS" : "Зараховано",
  "HISTBALL100" : "127,0"
}
```

Малюнок 19 Приклад об'єкту з колекції all0000

```
"data" : [
  "OUTID",
  "BIRTH",
  "SEXTYPENAME",
  "REGNAME",
  "AREANAME",
  "TERNAME",
  "REGTYPENAME",
  "TERTYPENAME",
  "CLASSPROFILENAME",
  "CLASSLANGNAME",
  "EONAME",
  "EOTYPENAME",
  "EOREGNAME",
  "EOAREANAME",
  "EOTERNAME",
  "EOPARENT",
  "UKRTEST",
  "UKRTESTSTATUS",
  "UKRBALL100",
```

Малюнок 20 Частина масиву назв стовпчиків із колекції headers0000

```
{
  "_id" : ObjectId("5eb69dd75d3fbc4e543234ee"),
  "name" : "REGTYPENAME",
  "data" : [
    "Випускник закладу загальної середньої освіти 2019 року",
    "Випускник минулих років",
    "Випускник, який здобуде в 2019 році повну загальну середню освіту в закордонному закладі освіти",
    "Студент закладу вищої освіти",
    "Учень (слухач) закладу професійної (професійно-технічної) освіти"
  ]
}
```

Малюнок 21 Приклад об'єкту 1 з колекції uniqueFields0000

```
{
  "_id" : ObjectId("5eb69dd65d3fbc4e543234ea"),
  "name" : "SEXTYPENAME",
  "data" : [
    "жіноча",
    "чоловіча"
  ]
}
```

Малюнок 22 Приклад об'єкту 2 з колекції uniqueFields0000

```
"Державний навчальний заклад Львівське вище професійне училище побутового обслуговування" : {
  "Українська мова і література" : {
    "count" : 156,
    "sum" : 14550,
    "threshold" : 33,
    "absent" : 2
  },
  "Історія України" : {
    "count" : 153,
    "sum" : 14221,
    "threshold" : 35,
    "absent" : 2
  },
  "Англійська мова" : {
    "count" : 3,
    "absent" : 2,
    "threshold" : 1
  },
  "Математика" : {
    "count" : 6,
    "sum" : 602,
    "threshold" : 1
  }
},
```

Малюнок 23 Приклад об'єктів найнижчого рівня з колекції statsBySchool0000