

**ТАЛАШ ВЛАДИСЛАВ ГЕННАДІЙОВИЧ**

Допускається до захисту:

завідувач кафедри

інформаційних технологій,

д. т. н., доцент

\_\_\_\_\_ Т. В. Нескородева

«\_\_\_\_\_» \_\_\_\_\_ 2022р.

**РОЗРОБКА ВЕБ-ДОДАТКУ АНАЛІЗУ ДЕМОГРАФІЧНИХ  
ПОКАЗНИКІВ КРАЇН ЄВРОПИ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (магістерська) робота

Науковий керівник:

Т. В. Січко, доцент кафедри

інформаційних технологій

к. т. н., доцент

\_\_\_\_\_

Оцінка: \_\_\_\_\_ / \_\_\_\_\_ /

(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: \_\_\_\_\_

(підпис)

Вінниця 2022

## АНОТАЦІЯ

**Талаш В. Г.** Розробка веб-додатку аналізу демографічних показників країн Європи. Спеціальність 122 «Комп'ютерні науки», Освітня програма «Комп'ютерні технології обробки даних (Data Science)». Донецький національний університет імені Василя Стуса, Вінниця, 2022.

У кваліфікаційній роботі досліджено: способи прогнозування демографічних показників на кілька років, архітектурні підходи реалізації веб додатків, бібліотеки для візуалізації даних. Показано: практичне застосування алгоритмів лінійної та поліноміальної регресії, процес розробки веб-додатку. Встановлено: закономірності залежності одних демографічних даних від інших, точності роботи лінійної та поліноміальної регресій при прогнозуванні різних показників.

Ключові слова: лінійна регресія, поліноміальна регресія, Sclearn, Highcharts, PostgreSQL, Rest Api, Python, C#.

75 ст., 2 табл., 45 рис., 50 джерел.

**Talash V.** Development of the web-application that analyses demographic of the European countries. Specialty 122 «Computer Science», Programme «Computer data processing technologies». Vasyl' Stus Donetsk National University, Vinnytsia, 2022.

In the work, the following are investigated: methods of forecasting demographic indicators for several years, architectural approaches of the implementation of web applications, libraries for data visualization. Shown: practical application of linear and polynomial regression algorithms, web application development process. Established: regularities of dependencies of some demographic data on others, accuracy of linear and polynomial regressions when forecasting various indicators.

Keywords: linear regression, polynomial regression, Sclearn, Highcharts, PostgreSQL, Rest Api, Python, C#.

Page 75. Tabl. 2. Fig. 45. 50 Src.

## ЗМІСТ

|  |    |
|--|----|
| ВСТУП.....   | 4  |
| РОЗДІЛ 1 .....   | 6  |
| ОГЛЯД ПИТАННЯ .....  | 6  |
| 1.1 Поняття демографії.....  | 6  |
| 2.1 Методи демографічного аналізу.....                                     | 6  |
| 1.3 Галузі які охоплює демографія.....                                     | 8  |
| 1.4 Важливість демографічного аналізу .....                                | 9  |
| 1.5 Існуючі програми які збирають інформацію про населення .....           | 10 |
| 1.4 Задачі роботи .....  | 19 |
| РОЗДІЛ 2 .....   | 21 |
| ПОСТАНОВКА ЗАДАЧІ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ПРОДУКТУ .....                    | 21 |
| 2.1 Опис вхідних даних .....   | 21 |
| 2.2 Опис майбутнього додатку .....   | 23 |
| 2.3 Архітектура програмного забезпечення.....                              | 24 |
| 2.4 Структура системи веб-додатку .....                                    | 28 |
| 2.5 База даних веб-додатку.....  | 30 |
| 2.6 Лінійна та поліноміальна регресії .....                                | 34 |
| РОЗДІЛ 3 .....   | 38 |
| ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ .....                                     | 38 |
| 3.1 Способи реалізації.....  | 38 |
| 3.2 Реалізація клієнтської програми.....                                   | 46 |
| 3.3 Реалізація серверної частини .....                                     | 54 |
| 3.4 Умови застосування додатку .....                                       | 59 |
| РОЗДІЛ 4 .....   | 60 |
| АНАЛІЗ ДАНИХ ЗА ДОПОМОГОЮ ДОДАТКУ .....                                    | 60 |
| 4.1 Графічний аналіз даних .....   | 60 |
| 4.2 Прогнозування даних за допомогою лінійної та поліноміальної регресії . | 63 |
| 4.3 Переваги та недоліки додатку.....                                      | 67 |
| ВИСНОВКИ.....  | 69 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ .....                             | 70 |

## ВСТУП

Демографія відіграє провідну роль в життєдіяльності людини. Ця наука охоплює різні сфери від показників народжуваності до хоббі та інтересів людини. Не дивно, що багато компаній, державних установ та вчених намагаються поглибити своє розуміння демографічної ситуації в різних країнах. Для бізнесу важливо знати, як буде розвиватись демографічна ситуація в країні і які області можуть приносити великий прибуток в майбутньому. Різні державні установи також відстежують та прогнозують демографічну ситуацію, щоб запобігти деяким кризам, які стосуються населення країни – перенаселення, голод, дефіцит тощо. Вчені потребують вивчення демографічних даних для подальшого аналізу та прогнозування. Провідну роль у питаннях демографії займають європейські країни, оскільки вони зацікавлені у вирішенні питань пов'язані з населенням, а також спостерігають, яким чином населення змінюється – народжується, обирає нову релігію, емігрує, іммігрує тощо.

На сьогоднішній день існує багато програм, які вивчають демографічну ситуацію, надають необхідні дані, а також, роблять прогнози, яким чином ситуація в країні може змінитись. Такі програми зазвичай охоплюють прогнозування лише декількох параметрів, а саме, прогноз того на яку кількість людей населення збільшиться чи зменшиться протягом наступних декількох років. Деякі програми, надають графіки того, як ситуація змінювалась протягом певного періоду часу, це може бути корисно, коли користувач працює з вже існуючим даними, але в даних додатках відсутня можливість будувати графіків, які враховують прогноз.

**Мета кваліфікаційної роботи** – створення веб-додатку за допомогою якого можна робити прогнози по зміні демографічної ситуації в країнах Європи, будувати графіки, робити прогнози.

**Об'єкт дослідження** – процес прогнозування демографічних показників країн Європи на наступні декілька років, процес створення графіків для аналізу демографічних даних.

**Предмет дослідження** – методи регресивного аналізу і машинного навчання (лінійна та поліноміальна регресія).

**Методи дослідження:** аналіз та вивчення програмних засобів та підходів для вирішення поставленої задачі.

**Апробація результатів дослідження.** Результати кваліфікаційної (магістерської) роботи апробовано на III-ій Міжнародній науково-практичній конференції «РОЗПОДІЛЕНІ ПРОГРАМНІ СИСТЕМИ І ТЕХНОЛОГІЇ», яка відбулась 24-25 листопада 2022 року на базі Київського національного університету будівництва і архітектури. Тези на тему: «Огляд програм для аналізу демографічних даних».

## РОЗДІЛ 1

### ОГЛЯД ПИТАННЯ

#### 1.1 Поняття демографії

Демографія належить до наук, що вивчають населення. Вона являє собою науку, що досліджує закономірності відтворення населення. Під відтворенням населення мають на увазі постійне відновлення населення в результаті природного руху (народжень і смертей), міграцій (пересувань людей з однієї території на іншу) і переходів людей з одних станів в інші.

Слово "демографія" утворене від грецьких демос – “народ” і графо – “пишу”, тобто буквально його можна перекласти як народоописання. Вперше термін "демографія" для позначення науки про відтворення населення застосував французький вчений А. Гійяр в роботі "Елементи статистики людини, або Порівняльна демографія" [1].

Демографія вивчає такі емпіричні дані про населення, як: кількість, статево-вікова структура, приріст (скорочення) населення та фізичне переміщення населення (міграція). До задач демографії належать: аналіз статистичних даних про рівні народжуваності і смертності, шлюбів і розлучень, тривалості життя; короткочасні та довгострокові прогнози стану та динаміки руху населення. Сім'я та шлюб — як соціальні інститути, соціалізація конкретно-історичних індивідів як масовий процес — також належать до комплексу об'єктів демографії. Основною ціллю демографії можна вважати виведення закономірностей та законів розвитку людності [2].

#### 2.1 Методи демографічного аналізу

Методи демографічного аналізу постійно оновлюються й розширюються у зв'язку зі зростанням вимог до точності обробки вихідної інформації про процеси відтворення населення, до строгості вимірювальних процедур, поняттєвого апарату й термінології, до адекватності використовуваних для аналізу й прогнозу демографічних моделей [3].

Низка проблем пов'язаних з вирішенням задач демографічного аналізу є досі нерозв'язаною. Наприклад, проблема взаємозв'язку реальних демографічних процесів та їх відображення системою демографічних показників. Також є необхідність у розвитку методологічних принципів перспективного обчислення населення.

Як і будь-яка інша наука, демографія має певний комплекс використовуваних нею методів. Сюди входять як загальнонаукові (аналіз, синтез, порівняння, узагальнення та ін.), так і спеціальні методи демографії і деяких суміжних з нею наук.

Демографічні методи дослідження покликані сприяти розкриттю специфіки відтворення населення, виявленню взаємозв'язку різних демографічних характеристик. До них, зокрема, відносяться методи подовжнього і поперечного аналізу.

До середини ХХ сторіччя поперечний аналіз був найрозповсюдженішим методом, він не втратив свого значення й сьогодні.

Про подовжній аналіз як про спосіб дослідження сукупності осіб, що водночас увійшли в певний демографічний стан, уявлення було сформульоване 1875 року В. Лексисом. Стосовно шлюбності реальних когорт підхід був використаний Р.Беком 1885 року при побудові таблиці припинення шлюбу. Перші спроби вивчення народжуваності у різних когортах були здійснені 1934 року англійським статистиком Д. Ричем.

В загальному: поперечний аналіз - є найпоширенішим прийомом демографічного аналізу. Багато які показники, з якими сьогодні має справу демографія, розраховують для умовної генерації. Велика частка окремих демографічних характеристик для року - шерехи вікових коефіцієнтів народжуваності, смертності, шлюбності, показники демографічних таблиць та сумарний коефіцієнт народжуваності, середня тривалість майбутнього життя. Все це дає уявлення про сучасний стан демографічних показників і збіг є тільки з показниками реальних генерацій лише у стабільному населенні.

Когортний метод або метод реальної генерації більш відомий як поздовжній аналіз, є способом вивчення демографічних процесів, за якого їх описують та аналізують у когортах, тобто сукупностях декількох людей, котрі ввійшли в якийсь демографічний стан. Когорта визначається демографічною подією та календарним періодом її формування. При цьому аналізі того чи іншого демографічного процесу періодом формування реальної когорти вважають період настання демографічної події, яка передувала тій події, частоту якої вимірюють. Для прикладу, при описанні розлучуваності у реальній шлюбній когорті розглядають сукупність осіб, які уклали шлюб у конкретному календарному періоді, та розлучення у цій когорті за період наступного шлюбного життя, розподіленні по тривалому шлюбу на момент кожного розлучення. Так, смертність гіпотетичної когорти 2006 року - це смертність усіх людей, котрі померли цього року, але народилися впродовж низки попередніх років.

### **1.3 Галузі які охоплює демографія**

Демографічні показники – це різні характеристики населення. Приклади демографічних показників можуть включати такі фактори, як раса, стать і вік населення, яке вивчається [4]. Статистична інформація про соціально-економічний стан населення відома як демографічні дані. Ці дані можуть містити інформацію про населення, наприклад:

- Віковий діапазон;
- расова та етнічна приналежність;
- стать;
- рівень освіти;
- дохід;
- зайнятість;
- заняття;
- власність житла;
- народжуваність;



- показники смертності;
- шлюбні показники;
- релігійна приналежність;
- політична приналежність;
- розмовна мова;
- географічне положення;
- хоббі та інтереси.

#### **1.4 Важливість демографічного аналізу**

Демографічні показники важливі, оскільки вони дають широке розуміння різних характеристик населення. Ця інформація особливо корисна державним організаціям для прийняття важливих політичних рішень щодо населення. Корпорації та інші підприємства в приватному секторі також використовують демографічні дані для кращого розуміння громадськості і, відповідно, складання стратегічних бізнес-планів і маркетингових планів [4].

Якщо ви займаєтесь професією, яка вимагає вивчення демографічних моделей, важливо зрозуміти нюанси.

У медицині та медичних дослідженнях, епідеміології та політиці охорони здоров'я та фармацевтиці моделі захворювань і протоколи лікування відрізняються залежно від демографічної когорти (вік, стать, географія, раса/етнічна приналежність, зайнятість тощо).

У маркетингових і ринкових дослідженнях, очевидно, точне націлювання на демографічні сегменти допомагає зрозуміти попит і переваги і, ймовірно, продавати більше.

Організації, що проводять опитування, звичайно, повинні розуміти демографію аж до дуже витонченого набору математичних рівнянь.

У різних державних сферах вам дійсно потрібно розуміти, як певна політика вплине на певні групи населення, від житла до освіти, охорони здоров'я і транспорту до кримінального правосуддя і безпеки праці, до підтримки літніх людей. Навіть податкова політика.

У сфері нерухомості корисно розповісти молодій парі, що поблизу є школи та лікарні, багато сімей по сусідству, культові споруди, які відповідають їхній релігійній вірі тощо, або порадити літньому клієнту, що там є хороші послуги для літніх людей, центр для людей похилого віку поблизу, легкий доступ до громадського транспорту та багато літніх сусідів для спілкування.

Фінансові організації також повинні добре володіти демографічними показниками, знати, як структурувати позики, які процентні ставки нараховувати, ймовірність дефолту, райони з найбільш стабільною вартістю нерухомості, на чому людям потрібно відкладати, пенсійні плани, і т. д. І, природно, страхові компанії покладаються на андерайтерів і актуаріїв для управління своїми ризиками.

Нарешті, на рівні загального інтересу та навчання ваших дітей будь-яка 9-річна дитина має знати, що Китай та Індія є двома найбільшими країнами світу, що в бідних країнах більше молодих людей, тоді як у багатих країнах населення старіє. Світ швидко зростає і потребує пошуку способів виробництва їжі та житла для підтримки своїх людей, старі люди мають різні захворювання, жінки мають інші проблеми, ніж чоловіки, тощо.

### **1.5 Існуючі програми які збирають інформацію про населення**

Програми, які збирають інформацію про населення є чудовим джерелом інформації для аналізу демографічних даних. Серед таких програм є веб-додаток Worldometer, який набув популярності під час пандемії COVID-19 (рис. 1.1) [5].

Worldometer, раніше Worldometers (множина) — довідковий веб-сайт, який надає лічильники та статистику в реальному часі з різних тем. Він належить та управляється компанією даних Dadaх, яка отримує дохід за рахунок онлайн-реклами.

W / Population

## Population

See also: [Demographics](#)

### World Population

World Population and top 20 Countries Live Clock. Population in the past, present, and future. Milestones. Global Growth Rate. World population by Region and by Religion. Population Density, Fertility Rate, Median Age, Migrants. All-time population total.

### Population by Country

Countries (and dependencies) in the world ranked by population. 1 Year change, migrants, median age, fertility rate, area (Km<sup>2</sup>), urban population, share of world's population, global rank. View also: [list of countries and dependencies](#) in two separate groups, and [most populous countries in 1950](#)

### Рисунок 1.1 – Сторінка сайту Worldometer

Сайт є частиною проекту статистики в реальному часі і, як стверджується, управляється «міжнародною командою розробників, дослідників та добровольців».

Сайт доступний 34 мовами і охоплює такі теми, як населення Землі, уряду, економіка, суспільство, ЗМІ, навколишнє середовище, їжа, вода, енергетика та охорона здоров'я.

У 2020 році Worldometer набув великої популярності завдяки розміщенню статистики, що відноситься до пандемії COVID-19.

Сайт має сторінку з таблицею де вказані дані про країни (рис. 1.2)

Search:

| #  | Country (or dependency)    | Population (2020) | Yearly Change | Net Change | Density (P/Km <sup>2</sup> ) | Land Area (Km <sup>2</sup> ) | Migrants (net) | Fert. Rate | Med. Age | Urban Pop % | World Share |
|----|----------------------------|-------------------|---------------|------------|------------------------------|------------------------------|----------------|------------|----------|-------------|-------------|
| 1  | <a href="#">Hungary</a>    | 9,660,351         | -0.25 %       | -24,328    | 107                          | 90,530                       | 6,000          | 1.5        | 43       | 72 %        | 0.12 %      |
| 2  | <a href="#">Austria</a>    | 9,006,398         | 0.57 %        | 51,296     | 109                          | 82,409                       | 65,000         | 1.5        | 43       | 57 %        | 0.12 %      |
| 3  | <a href="#">Germany</a>    | 83,783,942        | 0.32 %        | 266,897    | 240                          | 348,560                      | 543,822        | 1.6        | 46       | 76 %        | 1.07 %      |
| 4  | <a href="#">Bulgaria</a>   | 6,948,445         | -0.74 %       | -51,674    | 64                           | 108,560                      | -4,800         | 1.6        | 45       | 76 %        | 0.09 %      |
| 5  | <a href="#">France</a>     | 65,273,511        | 0.22 %        | 143,783    | 119                          | 547,557                      | 36,527         | 1.9        | 42       | 82 %        | 0.84 %      |
| 6  | <a href="#">Luxembourg</a> | 625,978           | 1.66 %        | 10,249     | 242                          | 2,590                        | 9,741          | 1.5        | 40       | 88 %        | 0.01 %      |
| 7  | <a href="#">Italy</a>      | 60,461,826        | -0.15 %       | -88,249    | 206                          | 294,140                      | 148,943        | 1.3        | 47       | 69 %        | 0.78 %      |
| 8  | <a href="#">Denmark</a>    | 5,792,202         | 0.35 %        | 20,326     | 137                          | 42,430                       | 15,200         | 1.8        | 42       | 88 %        | 0.07 %      |
| 9  | <a href="#">Finland</a>    | 5,540,720         | 0.15 %        | 8,564      | 18                           | 303,890                      | 14,000         | 1.5        | 43       | 86 %        | 0.07 %      |
| 10 | <a href="#">Slovakia</a>   | 5,459,642         | 0.05 %        | 2,629      | 114                          | 48,088                       | 1,485          | 1.5        | 41       | 54 %        | 0.07 %      |
| 11 | <a href="#">Ireland</a>    | 4,937,786         | 1.13 %        | 55,291     | 72                           | 68,890                       | 23,604         | 1.8        | 38       | 63 %        | 0.06 %      |
| 12 | <a href="#">Spain</a>      | 46,754,778        | 0.04 %        | 18,002     | 94                           | 498,800                      | 40,000         | 1.3        | 45       | 80 %        | 0.60 %      |
| 13 | <a href="#">Malta</a>      | 441,543           | 0.27 %        | 1,171      | 1,380                        | 320                          | 900            | 1.5        | 43       | 93 %        | 0.01 %      |
| 14 | <a href="#">Croatia</a>    | 4,105,267         | -0.61 %       | -25,037    | 73                           | 55,960                       | -8,001         | 1.4        | 44       | 58 %        | 0.05 %      |

Рисунок 1.2 – Сторінка з даними по країнах

Таблиця являє собою зручний спосіб представлення інформації, також тут знаходяться посилання на сторінки країни де можна отримати більш детальну інформацію щодо них. Сама сторінка містить інформацію про поточне населення країни та графік, що показує як населення змінювалось протягом останніх років. Також ця сторінка містить більш детальну інформацію про походження цих даних, та деякі дані які описують співвідношення населення даної країни до загального населення планети (рис 1.3).

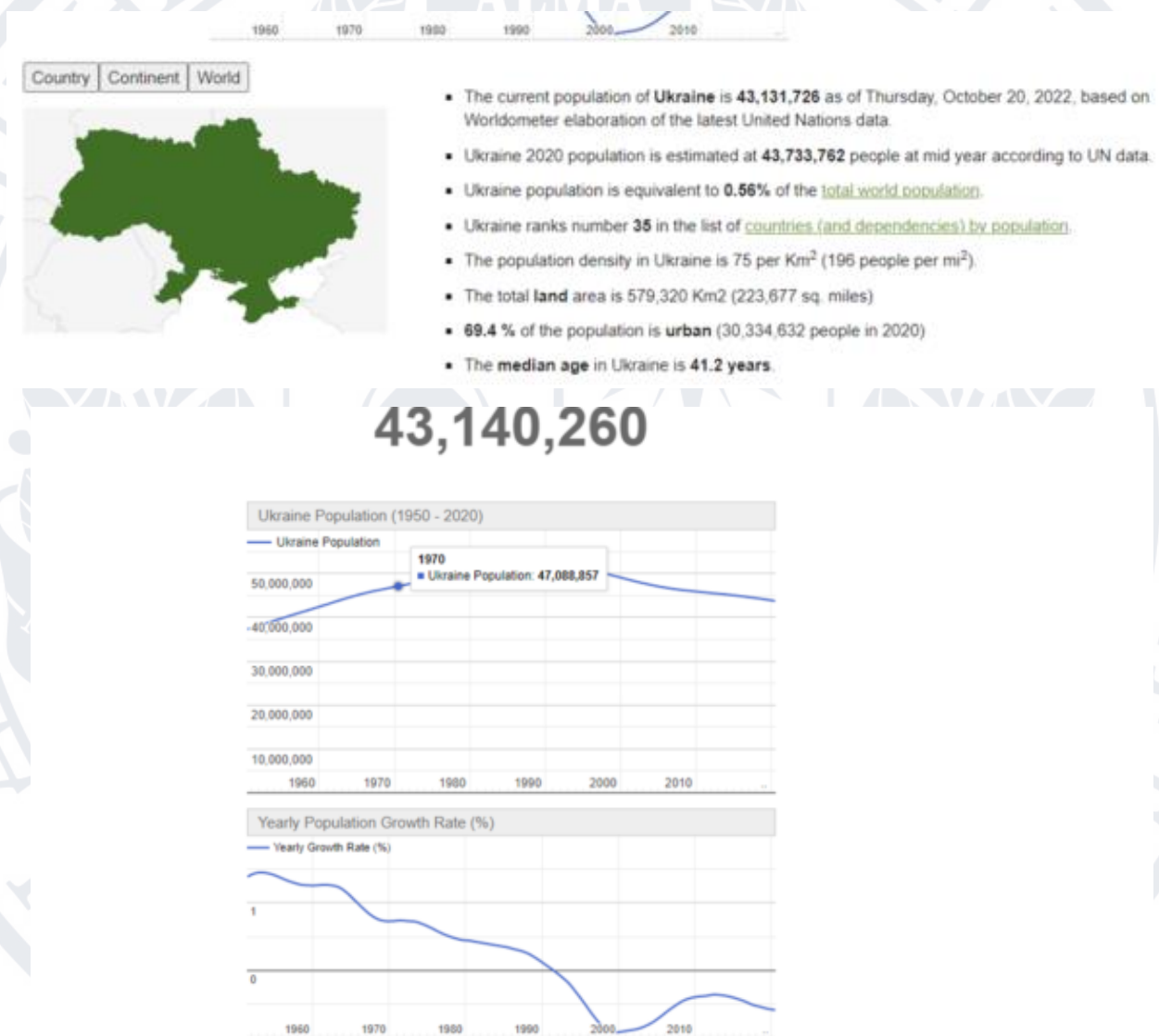


Рисунок 1.3 – Сторінка країни

Аналізуючи цей сайт можна виділити наступні особливості:

- Сайт містить багато даних про країни.
- Має зручний формат візуалізації даних.
- Недоліком є відсутність більш поглиблених даних про країну.

- Дані представлені тільки у форматі графіку.
- Немає можливості зробити прогноз, яким чином ситуація в країні може змінитись. Хоча є таблиці з прогнозами на наступні декілька років, що робиться за допомогою вбудованого алгоритму сайту.

Ще один сайт для слідкування за демографічною ситуацією це Worldpopulationreview (рис. 1.4).

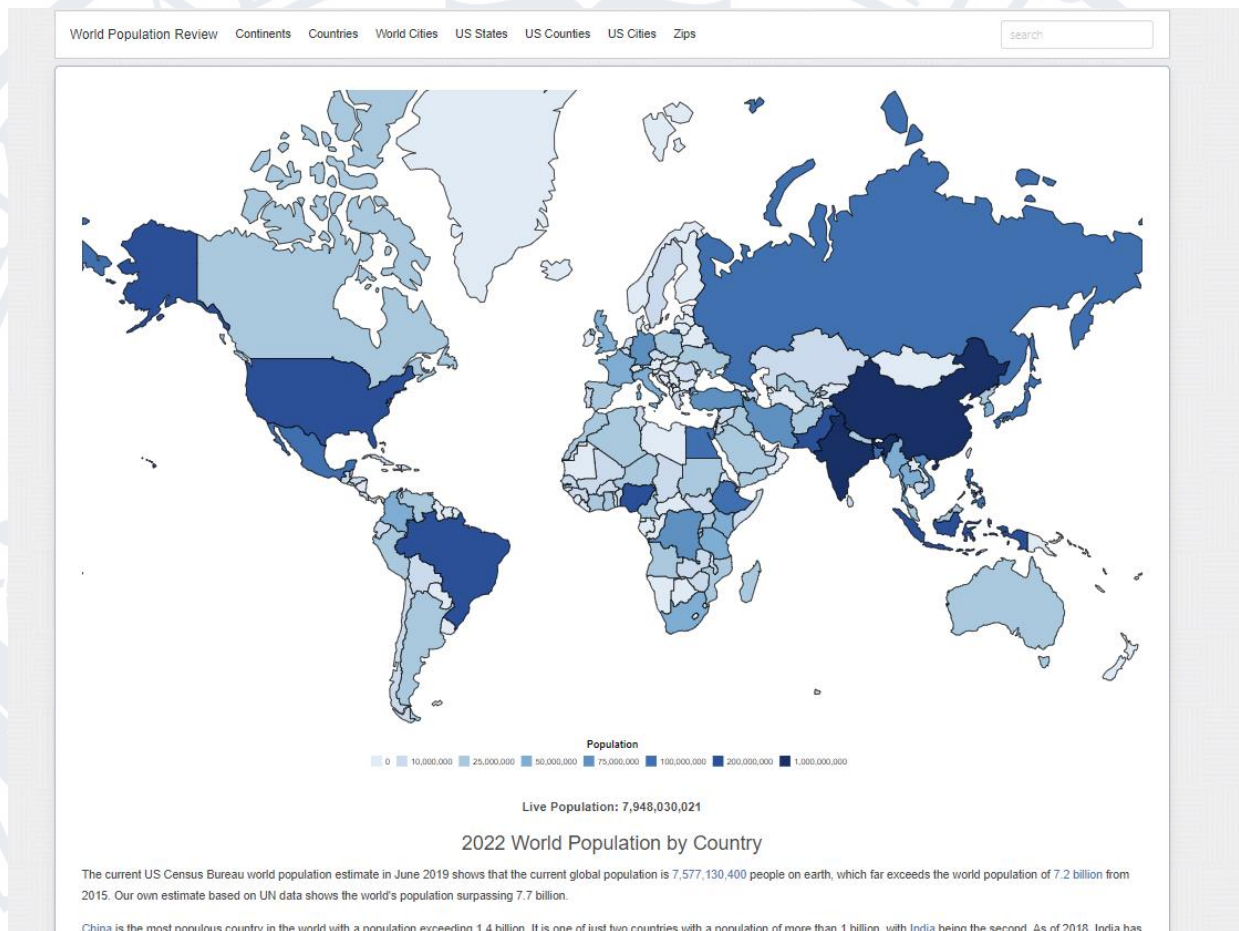


Рисунок 1.4 – Сторінка сайту Worldpopulationreview

World Population Review перетворює складну демографічну інформацію на зрозумілі статті про населення країн і міст.

Більшість демографічних даних приховано в електронних таблицях, за складними API або всередині громіздких інструментів. Мета World Population Review — зробити ці дані більш доступними за допомогою графіків, діаграм, аналізу та візуалізації [6].

World Population Review — незалежна організація без політичних поглядів.

Сайт має більш розгорнуті дані на рахунок статистики по країнах, є таблиці які містять, наприклад, країни розмір населення яких перевищує 100 000 000

людей. Міститься розгорнута інформація яка описує загалом демографічну ситуацію в тій чи іншій країні. Містить карту з країнами. Також на сайті є таблиця з переліком всіх країн світу та інформацію про населення кожної з них (рис. 1.5).

**2022 World Population 7,975,105,156**

Choose Year to Display Below:

↓ CSV ↕ JSON

| Flag | Country       | 2022 (Live) ▾ | 2020 Population | Area (km <sup>2</sup> ) | Density (km <sup>2</sup> ) | Growth Rate | World % | Rank |
|------|---------------|---------------|-----------------|-------------------------|----------------------------|-------------|---------|------|
|      | China         | 1,425,864,015 | 1,424,929,781   | 9,706,961               | 147/km <sup>2</sup>        | -0.00%      | 17.88%  | 1    |
|      | India         | 1,420,138,718 | 1,396,387,127   | 3,287,590               | 431/km <sup>2</sup>        | 0.68%       | 17.77%  | 2    |
|      | United States | 338,772,807   | 335,942,003     | 9,372,610               | 36/km <sup>2</sup>         | 0.38%       | 4.24%   | 3    |
|      | Indonesia     | 276,042,091   | 271,857,970     | 1,904,569               | 145/km <sup>2</sup>        | 0.64%       | 3.45%   | 4    |
|      | Pakistan      | 237,231,086   | 227,196,741     | 881,912                 | 267/km <sup>2</sup>        | 1.91%       | 2.96%   | 5    |
|      | Nigeria       | 220,137,692   | 208,327,405     | 923,768                 | 237/km <sup>2</sup>        | 2.41%       | 2.74%   | 6    |
|      | Brazil        | 215,612,217   | 213,196,304     | 8,515,767               | 25/km <sup>2</sup>         | 0.46%       | 2.70%   | 7    |
|      | Bangladesh    | 171,729,468   | 167,420,951     | 147,570                 | 1,160/km <sup>2</sup>      | 1.08%       | 2.15%   | 8    |
|      | Russia        | 144,701,597   | 145,617,329     | 17,098,242              | 8/km <sup>2</sup>          | -0.27%      | 1.81%   | 9    |

Рисунок 1.5 – Таблиця з даними

На кожну країну з цієї таблиці можна натиснути та опинитись на сторінці цієї країни. Сама сторінка країни є дуже схожою по своєму змісту з аналогом вище, хоча тут є інформація, яка більш детально розглядає демографічну ситуацію в цій країні (рис 1.6).

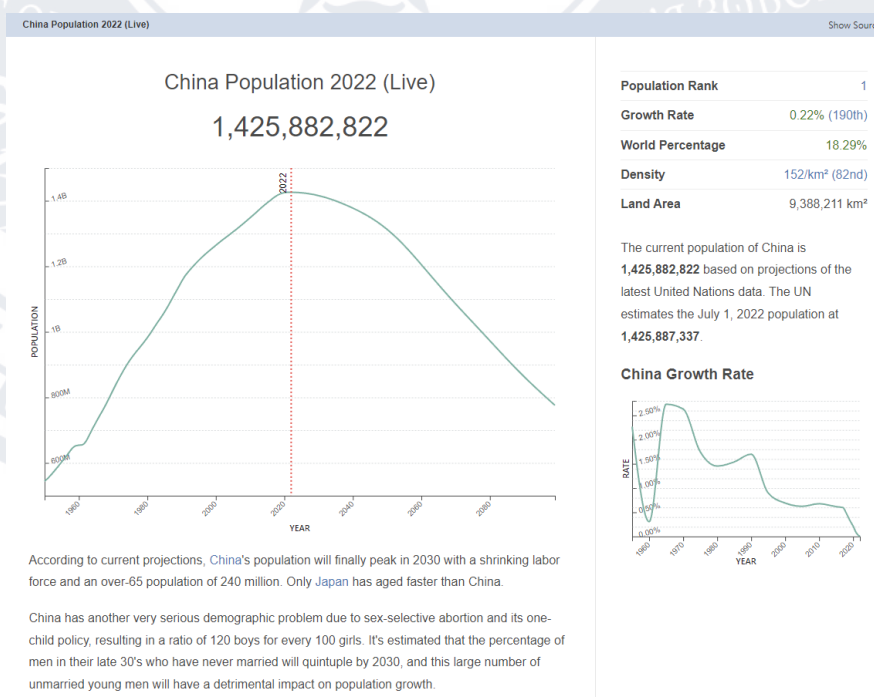


Рисунок 1.6 – Сторінка країни

Також на цьому сайті є «годинник населення», який дозволяє відраховувати кількість народжених людей за 15 хв та кількість померлих людей (рис. 1.7).

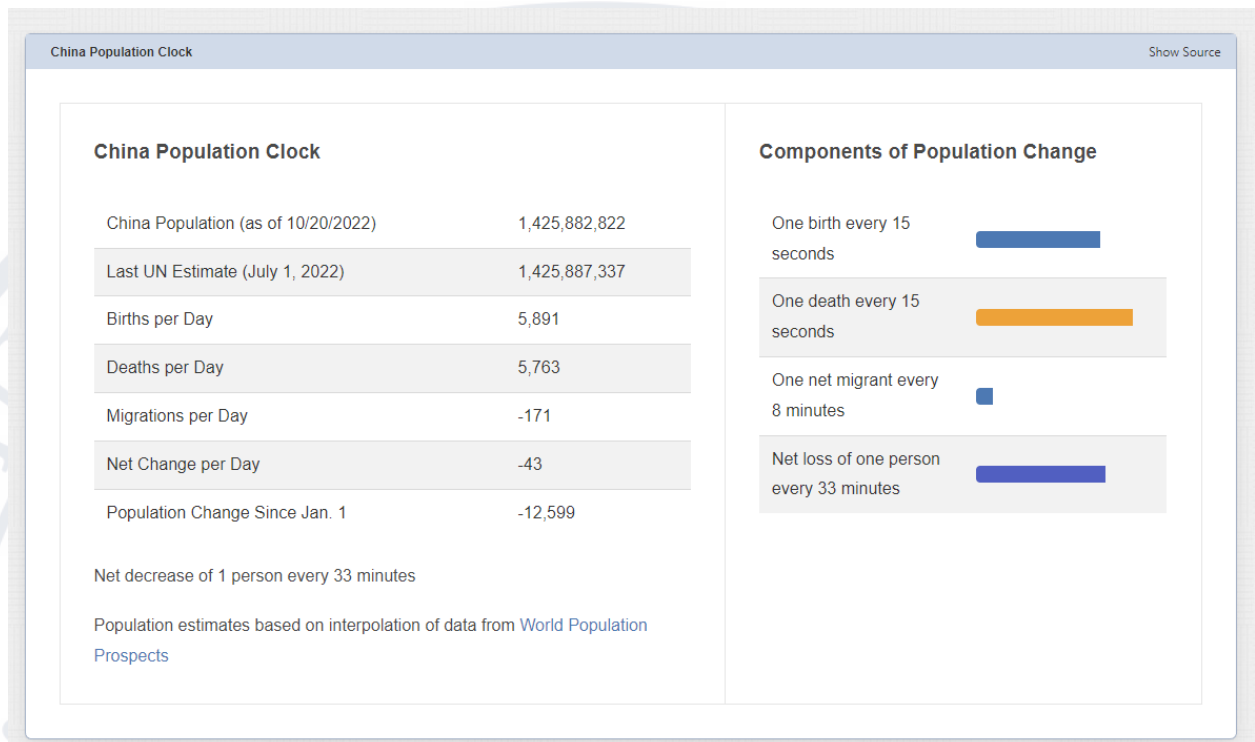


Рисунок 1.7 – «Годинник населення»

Аналізуючи цей сайт можна виділити наступні особливості:

- Сайт містить набагато більше даних про країни ніж аналог вище.
- Має зручний формат візуалізації даних.
- Має багато графічних візуалізацій, хоча немає візуалізації за допомогою карти.
- Немає можливості зробити прогноз, яким чином ситуація в країні може змінитись, ввівши власні дані, хоча має «годинник населення».

Наступний сайт, який збирає інформацію про населення – Ourworldindata. Our World in Data (OWID, з англ. «Наш світ у даних») — некомерційний електронний проект, що публікує у відкритому доступі дані про глобальні проблеми людства, такі як хвороби, голод, глобальне потепління, війни, катастрофи, соціальна нерівність, бідність . Також з кінця 2020 року портал компілює та систематизує світову статистику щодо пандемії COVID-19. Дані Our World in Data регулярно використовують провідні світові ЗМІ (наприклад, The

New York Times, The Economist, The Atlantic, The Guardian та інші), а дослідження - цитують у найбільших академічних журналах, включаючи Science, Nature, Proceedings of the National Academy of Sciences, The Lancet. Національні лідери, представники ООН та ВООЗ використовують матеріали видання на брифінгах. База даних із тестування на COVID-19 була опублікована в Nature.

Основні принципи проекту ґрунтуються на так званому «оптимістичному» підході до суспільного розвитку — інтелектуальної традиції, яскравим представником якої був шведський академік Ганс Рослінг, який використовує статистичні дані для ілюстрації позитивних трендів у світовому розвитку. Як і Рослінг, Розер вважає, що завдяки індустріалізації більшість людей значно зріс рівень життя, а масштаби таких труднощів, як голод, неписьменність і відсутність доступу до медицини, значно зменшилися. У 2015 році Розер навіть допомагав Рослінгу в підготовці документального фільму Бі-бі-сі Don't Panic - The Truth about Population (з англ. - "Не панікуйте - правда про населення") (рис. 1.8) [7].

The image shows a screenshot of the 'Our World in Data' website. The page features a dark blue header with the site's name, a search bar, and navigation links. Below the header is a yellow banner with the text 'COVID-19 vaccinations, cases, excess mortality, and much more' and a button to 'Explore our COVID-19 data'. The main content area has a light blue background and a large title 'World Population Growth' in a dark blue serif font. Below the title, it lists the authors: 'by Max Roser, Hannah Ritchie, Esteban Ortiz-Ospina and Lucas Rodés-Guirao' and notes the publication date: 'First published in 2013; most recent substantial revision in May 2019.' There are also links for 'Reuse our work freely' and 'Cite this research'.

When and why did the world population grow? And how does rapid population growth come to an end? These are the big questions that are central to this research article.

The world population increased from 1 billion in 1800 to around 8 billion today.

The world population growth rate declined from around 2% per year 50 years ago to under 1.0% per year.

Other relevant research:

Рисунок 1.8 – Сторінка сайту присвячена даним про світове населення



Сайт містить секцію про густоту населення кожної країни, дані представлені в декількох форматах, а саме:

- Таблиця;
- мапа;
- графік.

Також вказані джерела з яких були зібрані дані. Є можливість завантажити дані, та оперувати ними, як захоче користувач. Також є опис цієї секції про проблеми з якими може зіштовхнутись людство з перенаселенням (рис. 1.9).

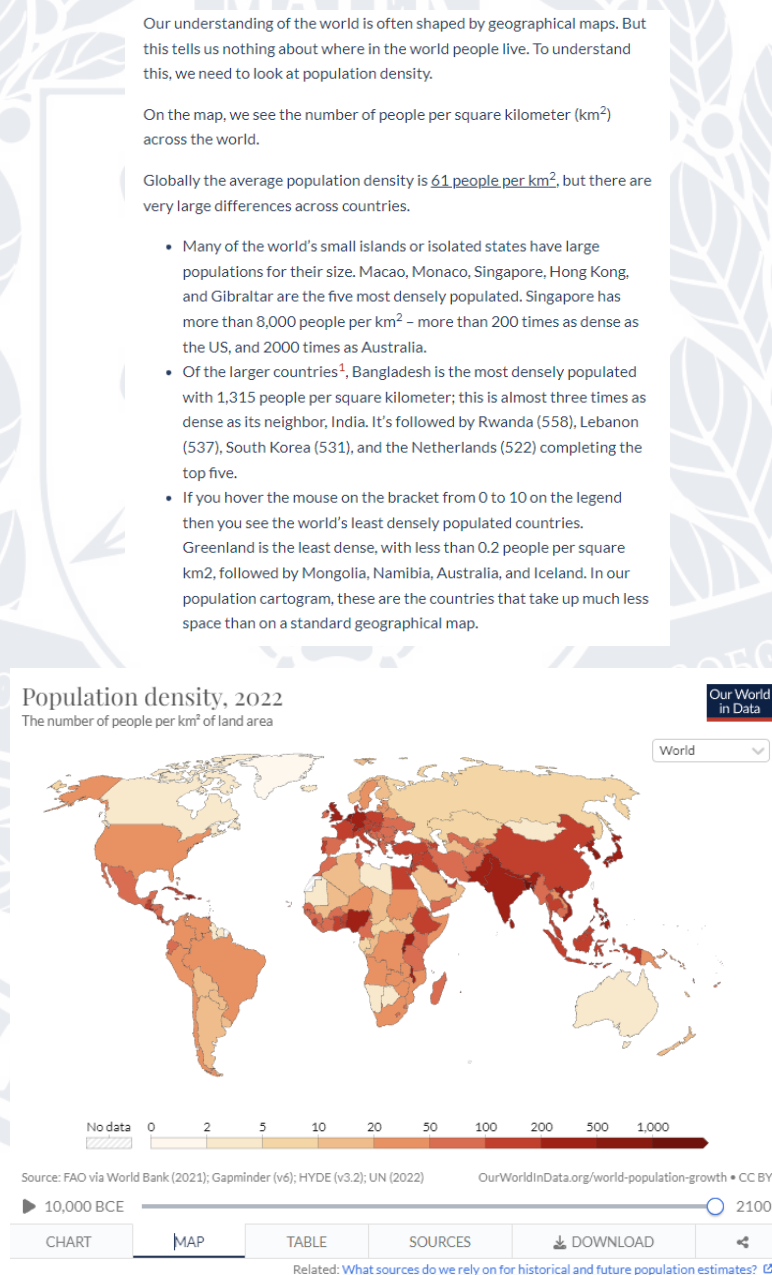
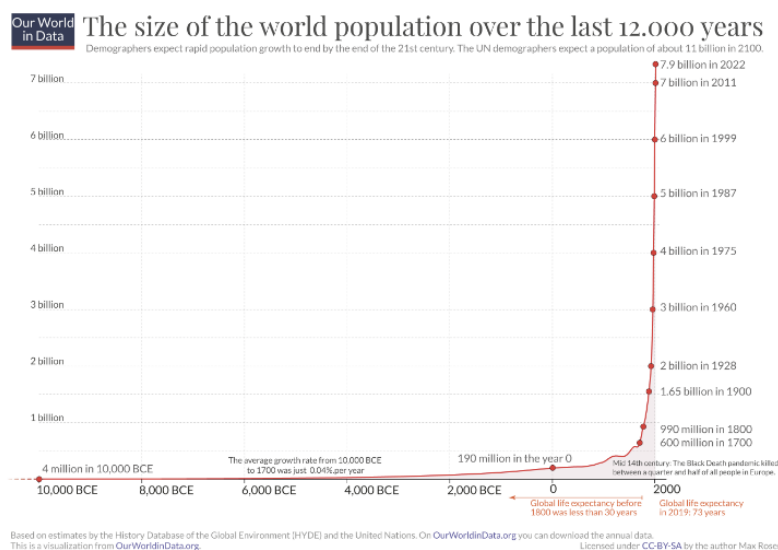


Рисунок 1.9 – Дані представлені у вигляді мапи та опис даних, що зображені на мапі

Також на сайті є секція, яка присвячена росту населення. На ній можна бачити опис проблеми, а також графік, що показує зміну населення планети протягом 10.000 років (рис. 1.10).



## World population from 10,000 BCE to today

The chart shows the increasing number of people living on our planet over the last 12,000 years. A mind-boggling change: the world population today is 1,860 times the size of what it was 12,000 ago when the world population was around 4 million – half of the current population of London.

What is striking about this chart is of course that almost all of this growth happened just very recently. Historical demographers estimate that around the year 1800 the world population was only around 1 billion people. This implies that on average the population grew very slowly over this long time from 10,000 BCE to 1700 (by 0.04% annually). After 1800 this changed fundamentally: the world population was around 1 billion in the year 1800 and has increased 7-fold since then.

Around 108 billion people have ever lived on our planet. This means that today's population size makes up 6.5% of the total number of people ever born.<sup>2</sup>

For the long period from the appearance of modern Homo sapiens up to the starting point of this chart in 10,000 BCE it is estimated that the total world population was often well under one million.<sup>3</sup> In this period our species was often seriously threatened by extinction.<sup>4</sup>

The interactive visualization is [here](#). And you can also download the [annual world population data](#) produced by Our World in Data.

A number of researchers have published estimates for the total world population over the long run, we have brought these estimates together and you can explore these various sources [here](#).

Рисунок 1.10 – Графік зростання населення планети та опис цього процесу.

Аналізуючи цей сайт можна виділити наступні особливості:

- Сайт містить інформацію в основному про зміну населення.
- Має зручний формат візуалізації даних.
- Має багато графічних візуалізацій.
- Немає можливості зробити прогноз, яким чином ситуація в країні може змінитись.
- Детальний опис в тій чи іншій секції.

Сайт більше спеціалізується на проблемах людства, ніж просто демографічному аналізу.

#### **1.4 Задачі роботи**

З огляду на вищесказані аналоги, можна зробити висновок, що сайти які мають дані про демографічні показники не мають функціоналу прогнозування зміни цих самих показників.

Метою даної роботи є створення веб-додатку за допомогою якого можна робити прогнози по зміні демографічної ситуації в країнах Європи.

Буде розроблятися веб-додаток саме для Європи, оскільки Європа є найбільш благополучною частина світу і користувачів може зацікавити дані про цей регіон, наступним фактором, який посприяв вибору саме Європи, це зацікавленість самих урядів європейських країн в аналізі демографічних показників їх країн, оскільки уряди європейських країн дуже ретельно відстежують зміни в демографії.

Розробку даної системи можна поділити на декілька етапів:

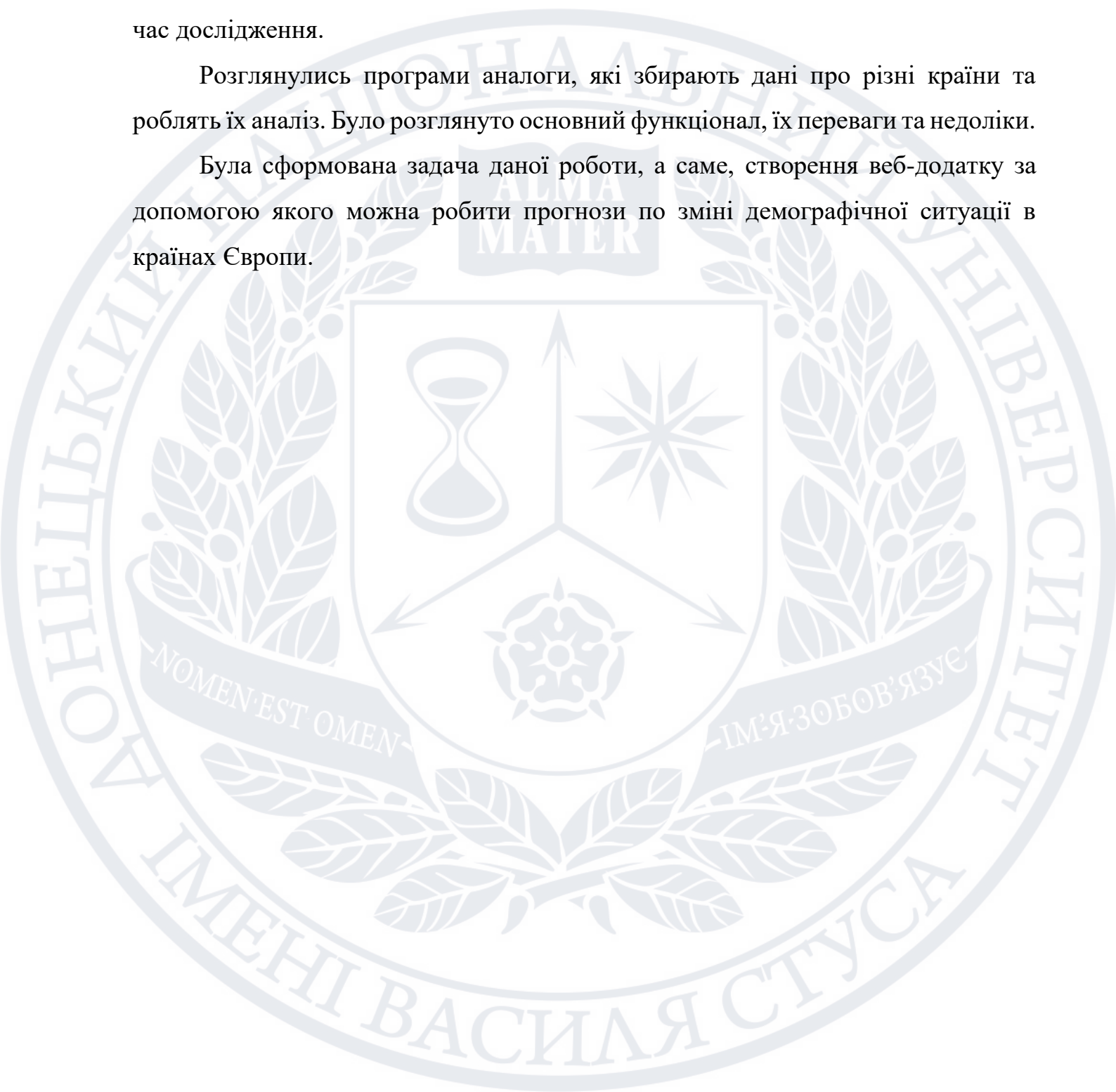
- Огляд архітектурних підходів, які дозволяють створити такий додаток.
- Проектування архітектури.
- Вибір та огляд технологій для реалізації.
- Реалізація клієнтської частини.
- Реалізація серверної частини.

#### **Висновки до розділу**

В даному розділі були розібрані поняття демографії, які задачі вона вирішує та яким чином може принести користь бізнесу, державним установам та вченим. Було визначено ключові демографічні показники, які будуть цікаві під час дослідження.

Розглянулись програми аналоги, які збирають дані про різні країни та роблять їх аналіз. Було розглянуто основний функціонал, їх переваги та недоліки.

Була сформована задача даної роботи, а саме, створення веб-додатку за допомогою якого можна робити прогнози по зміні демографічної ситуації в країнах Європи.



## РОЗДІЛ 2

### ПОСТАНОВКА ЗАДАЧІ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ПРОДУКТУ

#### 2.1 Опис вхідних даних

Метою даного веб-додатку є збирання та обробка інформації про демографічну ситуацію в країнах Європи. Оскільки демографія має багато значень, що можна аналізувати, було обрано основні демографічні показники, які цікавлять користувачів даної системи:

- Середня тривалість життя (Очікувана тривалість життя);
- стать;
- дохід;
- етнічна група;
- зайнятість;
- рівень освіти;
- густота населення.

Середня тривалість життя – показник, що вимірюється в роках та є ключовим показником здоров'я населення. Він має ширше значення ніж показник дитячої смертності, оскільки очікувана тривалість життя охоплює смертність протягом усього життя.

Стать – показник, що являє собою відсоток людей тієї чи іншої статі, в місті, області чи країні загалом.

Дохід – показник, що вимірюється в доларах та відповідає за середнє значення доходу в країні, області чи місті.

Етнічна група – показник, який є відсотком людей різних етнічних груп, представники інших країн або люди, які являють собою етнічну більшість в країні.

Зайнятість – відображає відсоток людей, який займається тією чи іншою професією.

Рівень освіти – буде відображати відсоток людей, які здобули ту чи іншу освіту: початкова освіта, базова середня освіта, повна середня освіта тощо.

Густота населення – показник, що відображає рівень заселеності певної території.

Окрім основних показників, які будуть відображатись користувачам даного веб-додатку, ще є показники країн, які можуть впливати на дані показники. Такі побічні дані можуть бути використані в передбачені змін того чи іншого показника.

- Для статі:
  1. Кількість людей тієї чи іншої статі за один рік в цілому – якщо сам показник статі відповідає за відсоток людей тієї чи іншої статі, що проживають на території країни в даний момент то цей показник показує тільки кількість людей певної статі, що народились за рік.
  2. Імміграція (стать) – показник, який відповідає за кількість людей, що прибули в країну і належать до певної статі.
- Для густоти населення:
  1. Кількість людей – популяція людей, яка є на даній території.
  2. Площа – площа певної адміністративної одиниці.
- Для етнічної групи:
  1. Імміграція (етнічна група) – кількість людей певної етнічної групи, що прибули до країни.
  2. Народжуваність – показник для відображення кількості народжених людей певної етнічної групи, які народились протягом певного року.
- Для середньої тривалості життя:
  1. Індекс охорони здоров'я – цей індекс дозволяє визначити, чи достатньо забезпечена країна медичними послугами, їх доступність, тощо.

2. Середній дохід в країні – показник, який також впливає на середню тривалість життя, оскільки забезпечені люди зможуть дозволити собі багато медичних послуг.

- Для освіти:

1. Індекс освіти – індекс освіти вимірюється рівнем грамотності дорослого населення (зі зважуванням у дві третини) та сукупним коефіцієнтом охоплення початковою, середньою та вищою школами (з зважуванням у одну третину).

Всі ці дані допоможуть користувачам проаналізувати демографічну ситуацію в тій чи іншій європейській країні а також зробити передбачення, того, як вона може змінюватись в подальших роках.

## 2.2 Опис майбутнього додатку

Необхідно розробити програмне забезпечення, яке дозволяє аналізувати демографічні дані в країнах Європи. Веб-додаток має мати наступний функціонал:

1. Збирати дані, про демографію країн Європи.
2. Мати візуальне відображення інформації, яке буде зрозуміле користувачеві.
3. За допомогою алгоритмів лінійної та поліноміальної регресії робити передбачення основних показників на наступний рік.
4. Виводити відсотки точності роботи алгоритмів.

Веб-додаток буде працювати на сервері. Є 2 види користувачів, що будуть користуватись додатком: звичайний користувач та адміністратор.

Функціонал для звичайного користувача:

- Відображення даних у вигляді таблиці, а також на карті.
- Можливість користувачеві заповнювати дані, що впливатимуть на той чи інший основний показник (але не зберігати введені ним дані в базі даних).

- На основі заповнених даних робити передбачення щодо основних демографічних показників.
- Візуалізація передбачених даних на наступний рік.
- Експорт даних в XLS, XLSX, CSV формати.

Функціонал для адміністратора:

- Можливість запуску веб-скрапінгу, для отримання даних.
- Можливість ручного введення даних.
- Можливість завантаження даних з Excel файлу.
- Користуватись веб-додатком як і звичайний користувач.

### 2.3 Архітектура програмного забезпечення

Дану систему можна поділити на наступні складові:

- Клієнт – програма, що буде виводити дані для користувача або подальшої передачі даних на сервер.
- Сервер – містить в собі програми, що будуть працювати на різних портах та обробляти дані, які надав клієнт, або записувати їх в базу даних.

Для написання серверної частини було вирішено створити два RESTful API для обробки даних та подальшого запису в базу даних.

Representational State Transfer (REST) – це архітектурний стиль програмного забезпечення, який визначає набір правил, які використовуватимуться для створення веб-служб. Веб-служби, які відповідають архітектурному стилю REST, відомі як веб-служби RESTful. Це дозволяє запитуючим системам отримувати доступ до веб-ресурсів і маніпулювати ними за допомогою єдиного попередньо визначеного набору правил [8]. REST API — це простий і гнучкий спосіб доступу до веб-служб без будь-якої обробки. Технології REST зазвичай надають перевагу перед більш надійною технологією простого протоколу доступу до об'єктів (SOAP), оскільки REST використовує меншу пропускну здатність, простий та гнучкий, що робить його більш придатним для використання в Інтернеті. Він використовується для отримання



або надання певної інформації з веб-служби. Уся комунікація через REST API використовує лише запит HTTP (рис. 2.1).

Комунікація клієнта з RESTful API відбувається за допомогою HTTP протоколу. HTTP – hypertext transfer protocol, протокол прикладного рівня, що дозволяє доставляти веб-документи веб-браузеру (або іншому клієнту). Раніше протокол призначався для передачі тільки гіпертекстової розмітки, але згодом він був модифікований для передачі інших типів даних, таких як: JSON, XML, WML.

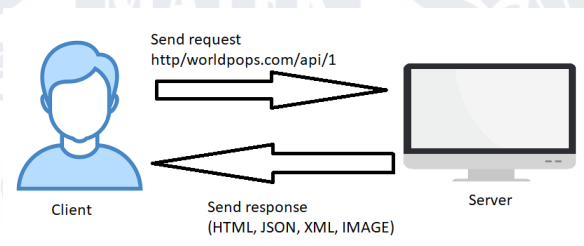


Рисунок 2.1 – Візуалізація взаємодії клієнта з REST-сервером

Є декілька принципів, яким притримується REST-архітектура:

- REST — це клієнт-серверна архітектура. Клієнт і сервер мають різні задачі. Сервер зберігає та/або обробляє інформацію та робить її доступною для користувача ефективним способом. Клієнт бере цю інформацію та відображає її користувачеві та/або використовує для виконання наступних запитів інформації. Такий розподіл завдань дозволяє як клієнту, так і серверу розвиватися незалежно, оскільки для цього потрібно лише, щоб інтерфейс залишався незмінним.
- REST не має статусу. Це означає, що зв'язок між клієнтом і сервером завжди містить усю інформацію, необхідну для виконання запиту. На сервері немає стану сеансу, він повністю зберігається на стороні клієнта. Якщо доступ до ресурсу вимагає автентифікації, то клієнт повинен аутентифікувати себе під час кожного запиту.
- REST можна використовувати кеш. Клієнт, сервер і будь-які проміжні компоненти можуть кешувати ресурси для підвищення продуктивності.
- REST додатково надає код на вимогу. Код можна завантажити, щоб розширити функціональність клієнта. Однак це необов'язково, оскільки

клієнт може не мати змоги завантажити або запустити цей код, і тому компонент REST не може покладатися на його виконання [9].

Вибір підходу побудови веб-додатку на користь RESTful API дозволяє клієнту звертатись до декількох API, таким чином ми можемо використовувати різні мови програмування для написання серверної частини веб-додатку. Комбінація різних мов програмувань дозволить ефективно використовувати переваги кожної з них, таким чином ми можемо не прив'язуватись до можливостей тільки одної мови програмування у вирішенні поставлених задач. Також цей підхід дозволяє використовувати різних клієнтів, отже ми можемо давати доступ до даних не тільки через клієнт веб-браузера, а і, наприклад, через мобільний додаток. Це доволі гнучкий підхід, особливо, коли додаток буде розвиватись та встановлюватись на різні платформи.

При написанні RESTful API можна використовувати багато архітектурних рішень. Серед всіх архітектурних рішень найбільш зручним та простим є багаторівнева архітектура. Багаторівнева архітектура поділяє програмне забезпечення на декілька шарів, серед них – шар представлення даних, шар бізнес-логіки програми, шар для доступу до бази даних (рис. 2.2) [10].



Рисунок 2.2 – Представлення багаторівневої архітектури

Шар представлення даних – це той рівень, з яким безпосередньо взаємодіє користувач. Цей рівень включає компоненти інтерфейсу користувача, механізм отримання введення від користувача. На даному рівні розташовані представлення і всі ті компоненти, які складають інтерфейс користувача – моделі представлень, контролери, об'єкти контексту запиту.

Шар бізнес-логіки – містить набори компонентів, які відповідають за обробку отриманих від рівня представлення даних, реалізує всю необхідну логіку додатків, усі обчислення, взаємодіє з базою даних і передає рівень представлених результатів обробки.

Шар доступу до даних – зберігає моделі, що описують використовувані сутності, також розміщуються специфічні класи для роботи з різними технологіями доступу до даних, наприклад, клас контексту даних. Тут також зберігаються репозиторії, якими рівень бізнес-логіки взаємодіє з базою даних.

Рівень доступу до даних не залежить від інших рівнів, рівень бізнес-логіки залежить від рівня доступу до даних, а рівень представлення - від рівня бізнес-логіки.

Компоненти, як правило, повинні мати слабкі зв'язки, тому невід'ємним атрибутом такої архітектури є процес використання залежностей.

При використанні такого архітектурного підходу ми можемо доволі швидко написати програму на відмінну від мікросервісного підходу. Хоча в майбутньому, при розширенні функціоналу програми, ми можемо зіштовхнутись з недоліком такого підходу, а саме, відсутність гнучкості при розширенні функціоналу.

При розробці сучасних веб-додатків також використовується таке поняття як фреймворк. Фреймворк у програмуванні — це інструмент, який надає готові компоненти або рішення, налаштовані для прискорення розробки. Фреймворк може включати бібліотеку, але визначається принципом інверсії керування (IoC). У традиційному програмуванні спеціальний код звертається до бібліотеки для доступу до багаторазового коду. За допомогою IoC фреймворк за потреби звертається до спеціальних фрагментів коду [11].

## 2.4 Структура системи веб-додатку

Система веб-додатку поділена на дві ролі: клієнт, адміністратор. Структурно систему було вирішено розділити на наступні компоненти серверу:

- AdminAuth – сервіс, що відповідає за автентифікацію адміністратора, перш ніж адміністратор зможе використовувати можливості системи. Сервіс при автентифікації видає користувачеві токен, який в подальшому буде використовуватись для запитів на сервер.
- AdminService – сервіс, що надає функціонал для адміністратора (заповнення бази даних даними про демографію). Для доступу до сервісу має використовуватись токен, який було згенеровано в AdminAuth сервісі, це робиться для того, щоб звичайний користувач не міг додавати дані в базу даних системи.
- ParseService – сервіс для запуску веб-парсеру який зможе збирати дані з різних сайтів та заповнювати ними бази даних, це один з трьох способів заповнення даними таблиць.
- ImportService – сервіс, що дозволяє імпортувати дані з файлу, який завантажив адміністратор системи. Файл має обов'язково бути правильного формату та містити всі необхідні колонки.
- AnalyzerService – сервіс, який надає функціонал для аналізу даних про демографію. В ньому знаходяться функції для використання лінійної та поліноміальної регресії, а також кластеризації даних.
- DataService – сервіс, що відповідає за надання даних користувачеві, він зможе віддавати дані по роках, які в свою чергу будуть поділятися на дані по країнах, по областях, по містам. Отже клієнт зможе вибрати будь який рік і конкретну локацію яка його цікавить.
- ExportService – сервіс, що дозволяє експортувати дані в формат CSV, XLS, та інші формати, що підтримуються програмою Excel. Таким чином користувач зможе імпортувати дані на свій пристрій і використовувати їх за власним бажанням.

Окрім серверної частини система складається з клієнтської частини. В свою чергу клієнтська частина поділяється на наступні компоненти:

- LoginComponent – компонент, для реалізації сторінки для автентифікації користувача на стороні клієнта.
- AdminComponent – компонент, для адміністратора системи, на ньому він зможе бачити дані, що знаходяться в базі даних, а також цей компонент надаватиме інтерфейс для виклику функціоналу для заповнення бази даних.
- HomeComponent – компонент, що доступний всім користувачам системи, на ньому буде представлено карту-діаграму з країнами Європи на які можна буде натиснути для більш детальну інформацію про кожну країну, також на цьому компоненті буде знаходитись таблиця з інформацією про кожну країну. В цьому компоненті є інтерфейс для надання системі даних для обробки та на основі них робити прогнози на наступний рік.
- CountryComponent – компонент, що являє собою карту певної країни з діаграмою про демографічні показники, також в цьому компоненті будуть вкладки з різними роками (максимум 5 років), натискаючи на які користувач може дізнатись демографічну ситуацію минулих років. Також в цьому компоненті є інтерфейс для надання системі даних для обробки та на основі них робити прогнози на наступний рік.
- RegionComponent, CityComponent – компоненти системи, які за своїм функціоналом дуже схожі на компонент CountryComponent, вони надають схожий інтерфейс для користувача і можуть робити те саме, що і компонент вище. Було обрано розбити систему саме так, оскільки в майбутньому ці компоненти можуть відрізнитись і також в цьому компоненті є інтерфейс для надання системі даних для обробки та на основі них робити прогнози на наступний рік тільки вже для міст чи області (будь-якої найбільшої адміністративної одиниці країни).

Загалом система матиме наступний вигляд (рис. 2.3):

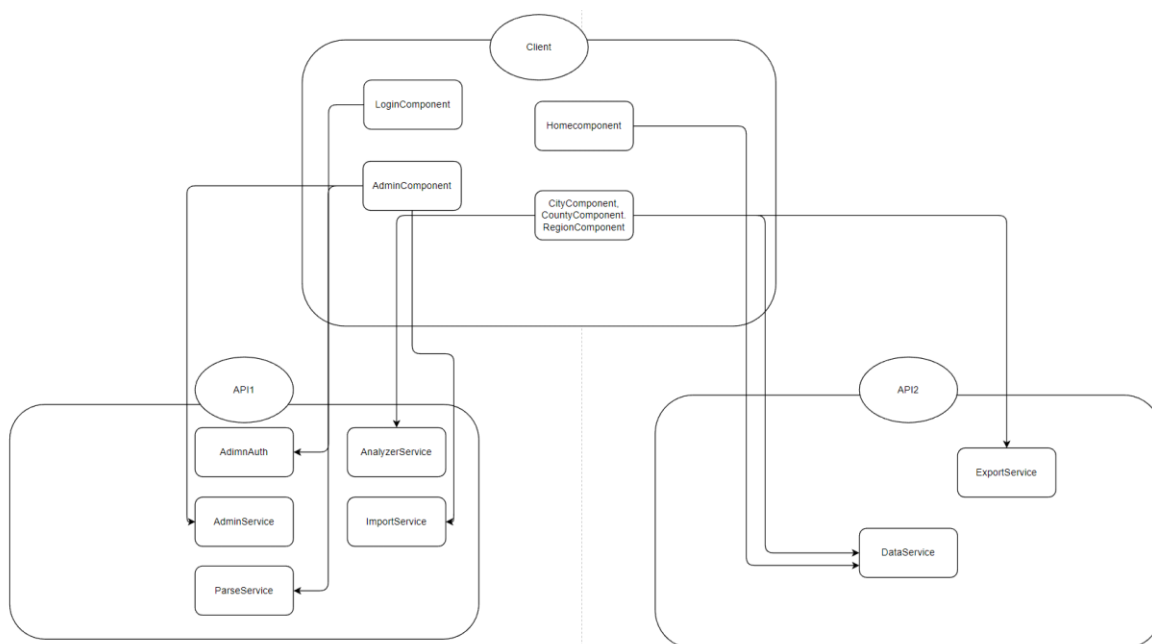
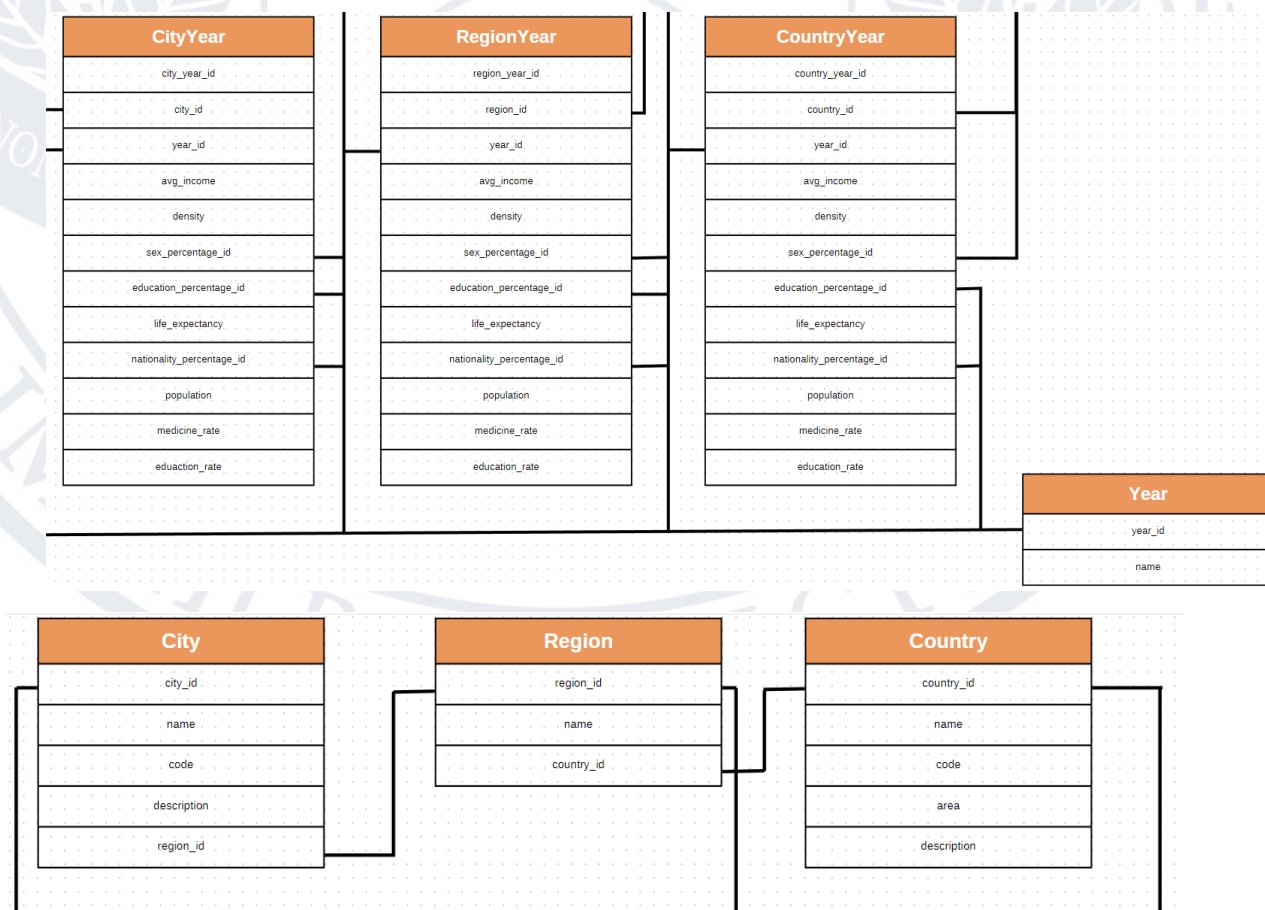


Рисунок 2.3 – Приблизна схема взаємодії компонентів

## 2.5 База даних веб-додатку

Загальна діаграма організації даних має виглядати наступним чином (рис.

2.4):



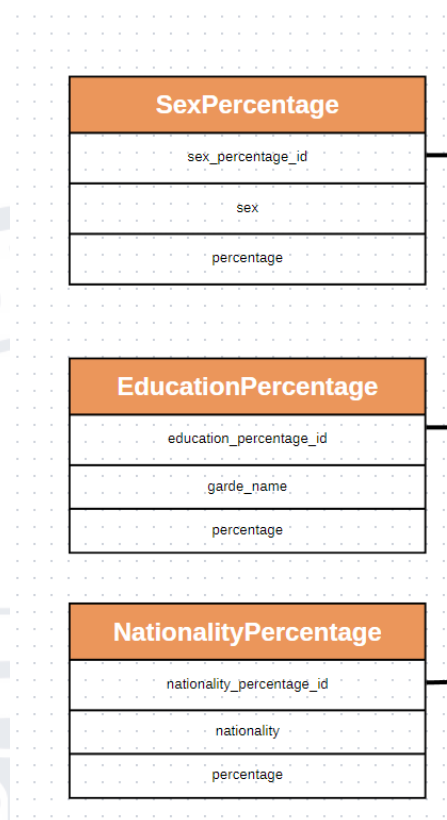


Рисунок 2.4 – Загальна діаграма організації даних

База даних має містити в собі всі необхідні дані, які пов'язані з демографією, також вона має бути гнучкою для подальшого розширення і додавання до неї нових полів та таблиць. Спроектвана база даних для веб-додатку для аналізу демографічних даних в країнах Європи буде мати наступний вигляд:

Таблиця «Country» - має інформацію про країну, вона має мати наступні поля:

- `country_id` – первинний ключ для таблиці, буде використовуватись для пошуку конкретної країни, а також `id` буде використовуватись як зовнішній ключ для інших таблиць.
- `name` – назва країни.
- `code` – код країни, має унікальне значення, тому може слугувати як обмеження при додаванні нової країни.
- `area` – площа країни.
- `description` – поле для короткого опису країни.

Таблиця «Region» - таблиця, що містить інформацію про найбільшу адміністративну одиницю країни. Ця таблиця матиме наступні поля:

- region\_id - первинний ключ для таблиці, буде використовуватись для пошуку конкретної країни, а також id буде використовуватись як зовнішній ключ для інших таблиць.
- name – назва регіону.
- country\_id – зовнішній ключ, який з'єднує таблиці «Country» та «Region».

Таблиця «City» - таблиця, що містить інформацію про місто. Ця таблиця матиме наступні поля:

- city\_id - первинний ключ для таблиці, буде використовуватись для пошуку конкретної країни, а також id буде використовуватись як зовнішній ключ для інших таблиць.
- name – назва міста.
- code – код міста, має унікальне значення, тому може слугувати як обмеження при додаванні нової країни.
- description – поле для короткого опису міста.
- region\_id - зовнішній ключ, який з'єднує таблиці «City» та «Region».

Таблиця «Year» - позначає рік. Ця таблиця матиме наступні поля:

- year\_id - первинний ключ для таблиці, буде використовуватись для пошуку конкретної країни, а також id буде використовуватись як зовнішній ключ для інших таблиць.
- name – число, що позначає рік (2015, 2017)

Наступні три таблиці мають схожу структуру, але в подальшому можуть бути різними. «CountryYear», «RegionYear», «CityYear» - проміжні таблиці в яких і зберігається основна інформація щодо демографії. Вони мають мати наступні поля:

- id – первинний ключ таблиці, в залежності яку таблицю ми використовуємо можуть мати різні імена, а саме: country\_year\_id, region\_year\_id, city\_year\_id.



- region\_id, city\_id, country\_id – зовнішні ключі, які мають відношення до різних таблиць, в залежності від назви буде використовуватись той чи інший ключ.
- year\_id – зовнішній ключ який відноситься до таблиці «Year».
- avg\_income – поле, яке відображає середній дохід населення.
- density – густина населення.
- sex\_percentage\_id – зовнішній ключ, що відноситься до таблиці «SexPercentage».
- education\_percentage\_id – зовнішній ключ, що відноситься до таблиці «EducationPercentage».
- life\_expectancy – очікувана тривалість життя (у роках).
- nationality\_percentage\_id – зовнішній ключ, що відноситься до таблиці «NationalityPercentage».
- population – кількість населення країни, міста чи регіону.
- medicine\_rate – індекс медицини.
- education\_rate – індекс освіти.

Таблиці «SexPercentage», «EducationPercentage», «NationalityPercentage» - будуть відображати інформацію про співвідношення певного критерію до відсотка, який цей критерій займає, в залежності від таблиці будуть наступні поля:

- sex\_percentage\_id – первинний ключ для таблиці, буде використовуватись для пошуку конкретної країни, а також id буде використовуватись як зовнішній ключ для інших таблиць.
- education\_percentage\_id – первинний ключ для таблиці, буде використовуватись для пошуку конкретної країни, а також id буде використовуватись як зовнішній ключ для інших таблиць.
- nationality\_percentage\_id – первинний ключ для таблиці, буде використовуватись для пошуку конкретної країни, а також id буде використовуватись як зовнішній ключ для інших таблиць.
- sex – стать.

- `garde_name` – назва ступеня освіти.
- `nationality` – назва національності.
- `percentage` – процент наявності того чи іншого показника.

## 2.6 Лінійна та поліноміальна регресії

Основним компонентом системи буде використання лінійної та поліноміальної регресії для прогнозування демографічної ситуації в наступні роки по будь яким показникам.

Лінійна регресія – це алгоритм машинного навчання, заснований на керованому навчанні. Він виконує завдання регресії. Регресія моделює цільове прогнозоване значення на основі незалежних змінних. Здебільшого використовується для з'ясування зв'язку між змінними та прогнозуванням. Різні моделі регресії відрізняються залежно від типу зв'язку між залежними та незалежними змінними, які вони розглядають, і кількості незалежних змінних, які використовуються (рис. 2.5) [12].

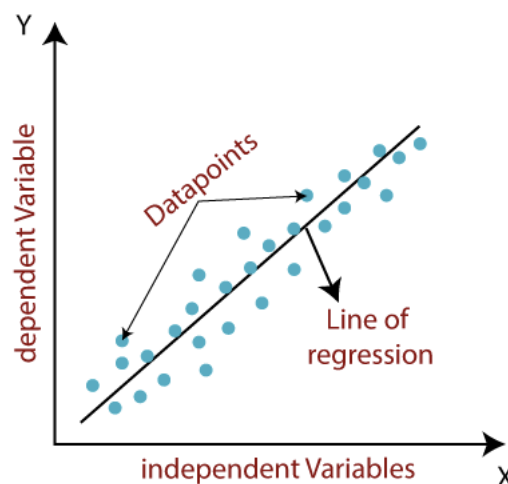


Рисунок 2.5 – Приклад лінійної регресії

Лінійна регресія виконує завдання прогнозування значення залежної змінної ( $y$ ) на основі заданої незалежної змінної ( $x$ ). Отже, ця техніка регресії визначає лінійну залежність між  $x$  (вхід) і  $y$  (вихід). Звідси й назва — лінійна регресія.

Лінійна регресія є одним із найпростіших і найпопулярніших алгоритмів машинного навчання. Це статистичний метод, який використовується для

прогнозного аналізу. Лінійна регресія робить прогнози для неперервних/дійсних або числових змінних, таких як продажі, зарплата, вік, ціна продукту тощо [13].

Лінійна регресія має багато практичних застосувань. Більшість програм належить до однієї з наступних двох широких категорій:

Якщо метою є передбачення, прогнозування або зменшення помилок, можна використати лінійну регресію, щоб адаптувати прогностичну модель до спостережуваного набору даних значень відповіді та пояснювальних змінних. Після розробки такої моделі, якщо зібрані додаткові значення пояснювальних змінних без відповідного значення відповіді, підібрану модель можна використовувати для прогнозування відповіді.

Якщо мета полягає в тому, щоб пояснити варіацію змінної відповіді, яку можна віднести до варіації пояснювальних змінних, лінійний регресійний аналіз можна застосувати для кількісної оцінки сили зв'язку між відповіддю та пояснювальними змінними, і, зокрема, для визначення того, чи деякі пояснювальні змінні можуть взагалі не мати лінійного зв'язку з відповіддю, або щоб визначити, які підмножини пояснювальних змінних можуть містити надлишкову інформацію про відповідь.

Моделі лінійної регресії часто підбираються за допомогою підходу найменших квадратів, але вони також можуть бути підігнані іншими способами [14].

Точність прогнозу за допомогою лінійної регресії також залежить від кількості «тренувальних» даних, так лінійна регресія, яка була побудована на даних за 60 років спостережень буде набагато точніше ніж модель побудована на 10 роках.

Алгоритм лінійної регресії працює лише тоді, коли зв'язок між даними є лінійним. Але припустимо, що якщо є нелінійні дані, то лінійна регресія не зможе намалювати найкращу лінію, і в таких умовах вона не працює. Є три причини використовувати поліноміальну регресію:

- Існують деякі зв'язки, які дослідники припускають як криволінійні. Очевидно, що такі типи випадків включатимуть поліноміальний термін.

- Перевірка залишків. Якщо спробувати адаптувати лінійну модель до викривлених даних, точкова діаграма залишків (вісь Y) на предикторі (вісь X) матиме плями з багатьох позитивних залишків посередині. Тому в такій ситуації це недоречно.
- У звичайному множинному лінійному регресійному аналізі припускається, що всі незалежні змінні є незалежними. У моделі поліноміальної регресії це припущення не виконується.

Поліноміальна регресія в основному використовуються для визначення або опису нелінійних явищ, таких як:

- Швидкість росту тканин.
- Прогресування епідемії захворювання.
- Розподіл ізотопів вуглецю в озерних відкладах.
- Може бути використана при прогнозуванні демографічної ситуації.

У статистиці, поліноміальна регресія є однією з форм регресійного аналізу, в якому залежність між незалежною змінною  $x$  і залежною змінною  $y$  моделюється як поліном від  $x$  ступеню  $n$ . Поліноміальна регресія відповідає нелінійній залежності між значенням  $x$  та відповідним умовним математичним сподіванням  $y$ , що позначається  $E(y | x)$ . Хоча поліноміальна регресія налаштовує нелінійній моделі даних, з боку теорії оцінювання ця задача є лінійною, в тому сенсі, що функція регресії  $E(y | x)$  є лінійною за невідомих параметрів які оцінюються за даними. З цього приводу поліноміальна регресія вважається приватним випадком множинної лінійної регресії (рис. 2.6) [15].

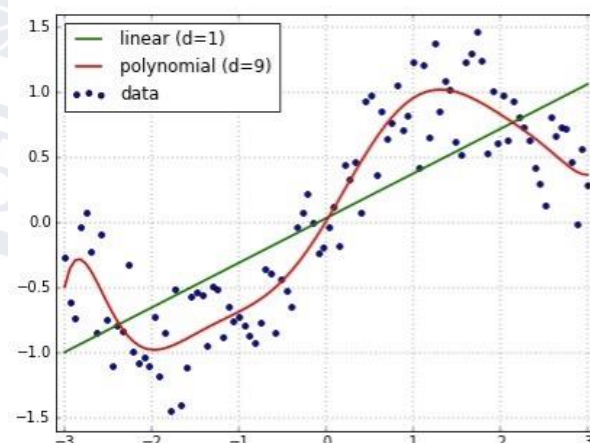


Рисунок 2.6 – Поліноміальна та лінійні регресії

Поліноміальна регресія має потенційно більшу точність ніж проста лінійна регресія, при побудові поліноміальної регресії потрібно обрати скільки кривих буде у кривої. Для цього підбирається поліном  $n$ -го ступеня.

Наведені вище методи машинного навчання допоможуть користувачам додатку при аналізі демографічної ситуації оскільки вони є методами прогнозування які базуються на великій кількості даних.

Лінійна та поліноміальна регресії можуть бути використані для наступних цілей.

- Прогнозування середню тривалість життя базуючись на даних про індекс здоров'я країни.
- Прогнозування середньої тривалості життя на основі середнього доходу в країні.
- Прогнозування середнього доходу базуючись на грамотності населення.

### **Висновки до розділу**

У даному розділі були розглянуті різні методи архітектури веб-додатків та було обрано модель REST API для побудови веб-додатку оскільки такий спосіб є сучасним та гнучким, у перспективі якщо потрібно розширити додаток, то це буде зробити легко через використання даного архітектурного підходу.

Під час проектування веб-додатку для аналізу демографічної ситуації в країнах Європи, компоненти системи було поділено на клієнтську та серверну частини, було описано взаємодію між всіма компонентами та спроектована майбутня база даних.

Було описано використання лінійної та поліноміальної регресій та наведено приклади, як ці методи машинного навчання можуть бути інтегровані в веб-додаток.

## РОЗДІЛ 3

### ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ

#### 3.1 Способи реалізації

Серверну частину додатку було реалізовано за допомогою двох мов програмувань, а саме: Python та C#.

Python — це високорівнева, універсальна та дуже популярна мова програмування. Мова програмування Python (останній Python 3) використовується у веб-розробці, програмах машинного навчання разом із усіма передовими технологіями в індустрії програмного забезпечення [16]. Мова програмування Python дуже добре підходить для початківців, а також для досвідчених програмістів з іншими мовами програмування, такими як C++ і Java. Наразі Python є найпоширенішою багатоцільовою мовою програмування високого рівня. Python дозволяє програмувати в об'єктно-орієнтованій і процедурній парадигмах. Програми Python зазвичай менші, ніж інші мови програмування, такі як Java [17]. Програмістам доводиться вводити відносно менше, а вимоги мови до відступів роблять їх читабельними весь час. Мова Python використовується майже всіма технологічними гігантами, такими як Google, Amazon, Facebook, Instagram, Dropbox, Uber... тощо.

Найбільшою перевагою Python є величезна колекція стандартної бібліотеки, яку можна використовувати для:

- Машинне навчання.
- Програми GUI (наприклад, Kivy, Tkinter, PyQt тощо).
- Веб-фреймворки, такі як Django (використовуються YouTube, Instagram, Dropbox).
- Обробка зображень (наприклад, OpenCV, Pillow).
- Веб-збирання (наприклад, Scrapy, BeautifulSoup, Selenium).
- Тестові фреймворки.
- Мультимедіа.
- Наукові обчислення.

- Обробка тексту та багато іншого..

Такі переваги, які надає Python дозволить швидко реалізувати необхідний функціонал додатку.

C# - об'єктно-орієнтована, компонентно-орієнтована мова програмування. C# надає мовні конструкції для безпосередньої підтримки цих концепцій, що робить C# природною мовою для створення та використання програмних компонентів. З моменту свого створення C# додав функції для підтримки нових робочих навантажень і новітніх практик проектування програмного забезпечення. За своєю суттю C# є об'єктно-орієнтованою мовою [18].

C# розроблено для роботи з платформою Microsoft .NET. Метою Microsoft є полегшення обміну інформацією та послугами через Інтернет, а також надання можливості розробникам створювати високопортативні програми. C# спрощує програмування завдяки використанню розширюваної мови розмітки (XML) і простого протоколу доступу до об'єктів (SOAP), які дозволяють отримати доступ до об'єкта або методу програмування, не вимагаючи від програміста писати додатковий код для кожного кроку. Оскільки програмісти можуть будувати на основі існуючого коду, а не повторно дублювати його, очікується, що C# дозволить швидше та дешевше виводити нові продукти та послуги на ринок [19].

Мова C# має наступні переваги:

- Швидший час розробки.
- Висока масштабованість.
- Об'єктно-орієнтована.
- Легка крива навчання.
- Велика спільнота розробників.
- Автоматичний збір сміття (пам'ять).
- Крос-платформеність.
- Зворотна сумісність.
- Краща цілісність і сумісність.

Загалом, C# є одною з найкращих сучасних мов програмування особливо, при побудові веб-додатків.

Для реалізації клієнтської частини додатку було використано фреймворк Angular.

Angular — написаний на TypeScript front-end фреймворк з відкритим кодом, який розробляється під керівництвом Angular Team [20]. Angular є частиною екосистеми JavaScript і одним із найпопулярніших інструментів розробки програмного забезпечення сьогодні. Він був представлений Google у 2009 році та отримав теплі рекомендації від спільноти розробників. Згідно з опитуванням StackOverflow 2022 року, 23 відсотки розробників програмного забезпечення застосовують фреймворк для створення інтерфейсів користувача [21].

Angular надає таку функціональність як двостороннє зв'язування, що дозволяє динамічно змінювати дані в одному місці інтерфейсу при зміні даних моделі в іншому, шаблони, маршрутизація і так далі [22]. (рис. 3.1).

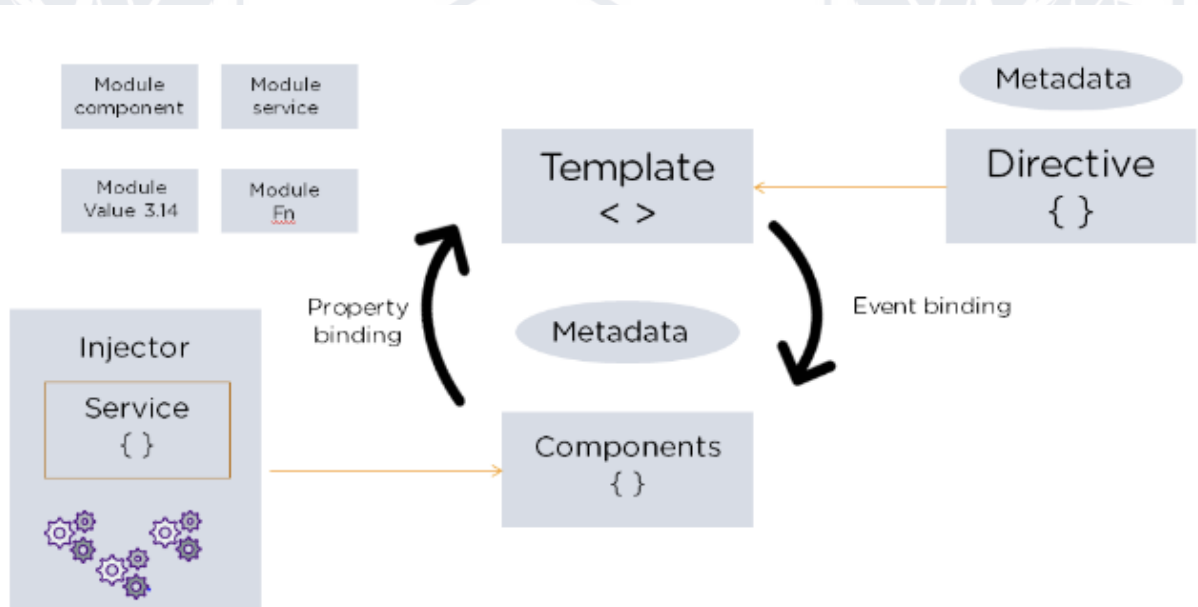


Рисунок 3.1 – Структура Angular-додатку, яка має в своїй основі вісім блоків.

Переваги використання фреймворку Angular:

- Власні компоненти.
- Прив'язка даних.



- Ін'єкція залежностей (DI).
- Тестування.
- Комплексний.
- Сумісність з браузером.

Для побудови RESTful API за допомогою мов Python та C# були використані наступні фреймворки:

Для API C# - ASP.NET – ASP.NET — це веб-фреймворк із відкритим кодом для створення веб-програм на основі .NET (dotNET). Він створений Microsoft, а версія 1.0 була випущена в 2002 році, щоб дозволити розробникам створювати динамічні веб-програми, служби та сайти. Фреймворк створено для роботи зі стандартним протоколом HTTP, який є стандартним протоколом, який використовується в усіх веб-додатках.

ASP.NET є наступником технології ASP (Active Server Pages) і є значним оновленням з точки зору гнучкості та потужності. Це розширення платформи .NET із додатковими інструментами та бібліотеками, призначеними спеціально для створення речей в Інтернеті, зокрема веб-програм і веб-сайтів [23].

Крім того, що нова платформа є кросплатформною, новий фреймворк взяв ASP.NET MVC, ASP.NET Web API і ASP.NET Web Pages і об'єднав їх в уніфікований MVC 6. Це дає розробникам вищу продуктивність і гнучкість, водночас отримуючи стабільну платформу, яка забезпечує безперебійну роботу їхніх програм і веб-сайтів.

Для Python був використаний фреймворк Django REST framework. Django REST framework – це потужний і гнучкий інструментарій для створення веб-інтерфейсів API [24].

Фреймворк Django REST базується на представленнях Django на основі класів, тому це чудовий варіант, якщо розробник вже знайомий з Django. Він використовує такі реалізації, як представлення на основі класів, форми, валідатор моделі, QuerySet тощо [25].

Для використання лінійної регресії, поліноміальної регресії, була використана бібліотека Python Scikit Learn. Scikit-learn - один із найбільш широко

використовуваних пакетів Python для Data Science і Machine Learning. Він дозволяє виконувати безліч операцій і надає безліч алгоритмів. Scikit-learn також пропонує відмінну документацію про свої класи, методи та функції, а також опис використовуваних алгоритмів [26].

Scikit-Learn підтримує:

- Попередню обробку даних;
- зменшення розмірності;
- вибір моделі;
- регресії;
- класифікації;
- кластерний аналіз.

Пакет Scikit-learn забезпечує наступні функції LinearRegression – для використання лінійної регресії. PolynomialFeatures – для використання поліноміальної регресії, поліноміальна регресія в бібліотеці не реалізована за замовчуванням, необхідно використовувати PolynomialFeatures, щоб можна було задавати кількість степенів в кривій.

Для веб-скрапінгу було використано бібліотеку BeautifulSoup — це бібліотека Python для отримання даних із файлів HTML і XML. Він працює з вашим улюбленим аналізатором, щоб забезпечити ідіоматичні способи навігації, пошуку та модифікації дерева аналізу. Зазвичай це економить години чи дні роботи програмістів [27].

Для зчитування даних з бази даних був використаний фреймворк Entity Framework Core. Entity Framework Core — це нова версія Entity Framework після EF 6.x. Це відкрита, легка, розширювана і кросплатформна версія технології доступу до даних Entity Framework.

Entity Framework — це структура об'єктного/реляційного відображення (O/RM). Це вдосконалення ADO.NET, яке надає розробникам автоматичний механізм для доступу та зберігання даних у базі даних [28].

EF Core призначений для використання з програмами .NET Core. Однак його також можна використовувати зі стандартними програмами на основі платформи .NET 4.5+ (рис. 3.2).

|                          |  |  |  |   |
|--------------------------|--|--|--|---|
| <b>Application Types</b> | ASP.NET Core Applications<br>Web, API, Console, etc. | .NET 4.5+ Applications<br>Console, WinForm, WPF, ASP.NET | Devices + IoT, Mobile, PC, Xbox, Surface Hub | Mobile Application<br>Android, iOS, Windows |
| <b>EF Core</b>           | EF Core  | EF Core  | EF Core                                      | EF Core                                     |
| <b>Framework</b>         | .NET Core  | .NET 4.5+  | UWP  | Xamarin                                     |
| <b>OS</b>                | Windows, Mac, Linux                                  | Windows  | Windows 10                                   | Mobile                                      |

Рисунок 3.2 - Підтримувані типи програм, .NET Frameworks і ОС.

Для побудови графічного інтерфейсу користувача використовувались компоненти які надавала бібліотека PrimeNG. PrimeNG — це багатий набір власних компонентів інтерфейсу Angular з відкритим кодом (рис. 3.3).

Дана бібліотека дозволяє використовувати вже готові дизайнерські рішення, які мають гарний вигляд, а також розробник може не відволікатись на процес створення власного дизайну для веб-додатку [29].

Для відображення інформації на картах, графіках тощо, була використана бібліотека Highcharts (рис. 3.4).

Бібліотека Highcharts містить усі інструменти, необхідні для створення надійних і безпечних візуалізацій даних. Бібліотеки діаграм, створені на основі JavaScript і TypeScript, працюють із будь-якою серверною базою даних або серверним стеком. Highcharts пропонує оболонки для найпопулярніших мов програмування (.Net, PHP, Python, R, Java), а також для iOS і Android, таких фреймворків, як Angular, Vue і React [30].

| List of Customers                          |              |                |            |             |             |                                 |  |
|--|--------------|----------------|------------|-------------|-------------|---------------------------------|--|
| Name                                       | Country      | Agent          | Date       | Balance     | Status      | Activity                        |  |
| <input type="checkbox"/> James Butt        | Algeria      | Iori Bowcher   | 09/13/2015 | \$70,663.00 | UNQUALIFIED | <div style="width: 10%;"></div> |  |
| <input type="checkbox"/> Josephine Darakyy | Egypt        | Amy Elsner     | 02/09/2019 | \$82,429.00 | PROPOSAL    | <div style="width: 10%;"></div> |  |
| <input type="checkbox"/> Art Venere        | Panama       | Asiya Javayant | 05/13/2017 | \$28,334.00 | QUALIFIED   | <div style="width: 20%;"></div> |  |
| <input type="checkbox"/> Lenna Paprocki    | Slovenia     | Xuxue Feng     | 09/15/2020 | \$88,521.00 | NEW         | <div style="width: 10%;"></div> |  |
| <input type="checkbox"/> Donette Foller    | South Africa | Asiya Javayant | 05/20/2016 | \$93,905.00 | PROPOSAL    | <div style="width: 10%;"></div> |  |
| <input type="checkbox"/> Simona Morasca    | Egypt        | Ivan Magalhaes | 02/16/2018 | \$50,041.00 | QUALIFIED   | <div style="width: 20%;"></div> |  |
| <input type="checkbox"/> Mitsue Tollner    | Paraguay     | Ivan Magalhaes | 02/19/2018 | \$58,706.00 | RENEWAL     | <div style="width: 20%;"></div> |  |
| <input type="checkbox"/> Leota Dillard     | Serbia       | Onyama Limba   | 08/13/2019 | \$26,640.00 | RENEWAL     | <div style="width: 20%;"></div> |  |
| <input type="checkbox"/> Sage Wieser       | Egypt        | Ivan Magalhaes | 11/21/2018 | \$65,369.00 | UNQUALIFIED | <div style="width: 10%;"></div> |  |
| <input type="checkbox"/> Kris Marner       | Mexico       | Onyama Limba   | 07/07/2015 | \$63,451.00 | PROPOSAL    | <div style="width: 10%;"></div> |  |

Showing 1 to 10 of 200 entries

Рисунок 3.3 – Приклад компоненту таблиці з бібліотеки PrimeNG.

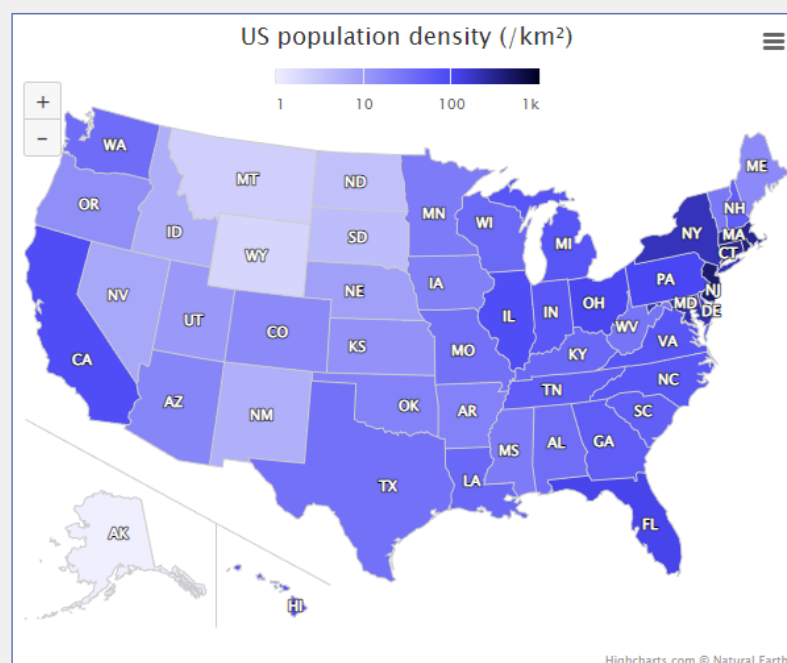


Рисунок 3.4 – Приклад використання бібліотеки Highcharts для позначення густоти населення в країні.

Для розробки клієнтської частини програми було використано редактор вихідного коду Visual Studio Code. Visual Studio Code — це легкий, але потужний редактор вихідного коду, який працює на робочому столі та доступний для Windows, macOS і Linux [31]. Серед переваг такого редактору, найголовнішою є те, що він підтримує JavaScript, TypeScript і Node.js. А також дає можливість

встановлення розширень для інших мов програмування таких як: C++, C#, Java, Python, PHP, Go, .NET.

Редактор зручний у користуванні за рахунок «легкості» інтерфейсу, а також поглибленого механізму кастомізації, тобто розробник може налаштувати редактор під свої потреби.

Для розробки веб-додатку на мовах програмування C# та Python були використані середовища розробки: Visual Studio та PyCharm відповідно.

Visual Studio IDE — це творчий стартовий майданчик, який можна використовувати для редагування, налагодження та створення коду, а потім для публікації програми. Окрім стандартного редактора та інструменту налаштування, які пропонують більшість IDE, Visual Studio містить компілятори, інструменти авто-завершення коду, графічні дизайнери та багато інших функцій для покращення процесу розробки програмного забезпечення [32].

PyCharm — це спеціальне інтегроване середовище розробки Python (IDE), яке надає широкий спектр основних інструментів для розробників Python, тісно інтегрованих для створення зручного середовища для продуктивної розробки на додатків різного типу [33].

Для реалізації бази даних було використано систему управління базами даних PostgreSQL.

PostgreSQL – потужна система управління базами даних з відкритим вихідним кодом. Ця система має декілька інструментів, таких як: сам сервер, графічний клієнт pgAdmin для перегляду баз даних, а також для зручного редагування запитів до БД [35].

PostgreSQL підтримує транзакції, підвибірки, тригери, перегляди, посилальну цілісність зовнішнього ключа та складне блокування. Він працює на багатьох платформах, включаючи Linux, більшість різновидів UNIX, Mac OS X, Solaris, Tru64 і Windows. Він підтримує текст, зображення, звуки та відео та включає інтерфейси програмування для C / C++, Java, Perl, Python, Ruby, Tcl і Open Database Connectivity (ODBC) [34].

### 3.2 Реалізація клієнтської програми

Для створення проекту на фреймворкові Angular було використано утиліту командної стрічки Angular CLI. За допомогою неї було створено основний каркас клієнтського додатку. Для створення додатку необхідно виконати наступну команду, яка вказана на рис. 3.5.

```
C:\Users\TVG\Desktop\masters-work\ng new eupopulation-client
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE eupopulation-client/angular.json (3117 bytes)
CREATE eupopulation-client/package.json (1083 bytes)
CREATE eupopulation-client/README.md (1064 bytes)
CREATE eupopulation-client/tsconfig.json (863 bytes)
CREATE eupopulation-client/.editorconfig (274 bytes)
CREATE eupopulation-client/.gitignore (620 bytes)
CREATE eupopulation-client/.browserslistrc (600 bytes)
CREATE eupopulation-client/karma.conf.js (1436 bytes)
CREATE eupopulation-client/tsconfig.app.json (287 bytes)
CREATE eupopulation-client/tsconfig.spec.json (333 bytes)
CREATE eupopulation-client/src/favicon.ico (948 bytes)
CREATE eupopulation-client/src/index.html (304 bytes)
CREATE eupopulation-client/src/main.ts (372 bytes)
CREATE eupopulation-client/src/polyfills.ts (2338 bytes)
CREATE eupopulation-client/src/styles.css (80 bytes)
CREATE eupopulation-client/src/test.ts (745 bytes)
CREATE eupopulation-client/src/assets/.gitkeep (0 bytes)
CREATE eupopulation-client/src/environments/environment.prod.ts (51 bytes)
CREATE eupopulation-client/src/environments/environment.ts (658 bytes)
CREATE eupopulation-client/src/app/app-routing.module.ts (245 bytes)
CREATE eupopulation-client/src/app/app.module.ts (393 bytes)
CREATE eupopulation-client/src/app/app.component.html (23364 bytes)
CREATE eupopulation-client/src/app/app.component.spec.ts (1112 bytes)
CREATE eupopulation-client/src/app/app.component.ts (223 bytes)
CREATE eupopulation-client/src/app/app.component.css (0 bytes)
| Installing packages (npm)...
```

Рисунок 3.5 – використання команди ng new для створення нового Angular-проекту

Ця команда запропонує інтегрувати систему маршрутизації в проект, а також можливість вибрати, якого формату будуть стилі, це може бути звичайний CSS або якийсь препроцесор.

Після створення даного проекту він буде мати базову структуру, у якій весь вихідний код програми буде знаходитись в директорії src (рис. 3.6).

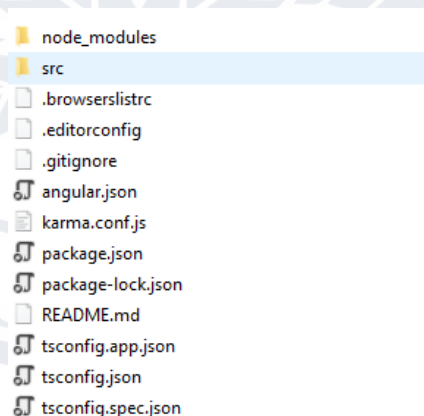


Рисунок 3.6 – базова структура Angular-проекту

Також утиліта Angular CLI дозволяє за допомогою команди `generate` створити необхідні компоненти, тільки потрібно вказати, що саме ми хочемо згенерувати (рис. 3.7).

```
C:\Users\TVG\Desktop\masters-work\euopulation-client>ng generate component Home
CREATE src/app/home/home.component.html (19 bytes)
CREATE src/app/home/home.component.spec.ts (612 bytes)
CREATE src/app/home/home.component.ts (267 bytes)
CREATE src/app/home/home.component.css (0 bytes)
UPDATE src/app/app.module.ts (467 bytes)
```

Рисунок 3.7 – використання команди `ng generate` для створення компоненту `HomeComponent`

Як видно з рис. 3.2.3 – ця команда створює декілька файлів, а саме `typescript`-файл у якому міститься логіка компоненту, файл з шаблоном HTML, файл зі стилями CSS (можливий інший формат, якщо розробник вибрав інший препроцесор для написання стилів) та файл тесту SPEC. Таким чином ця команда дозволяє швидко створити необхідні файли для написання компонентів, а також додає новостворений компонент у файл модулів, щоб компонент міг бути викликаний з інших компонентів. Таким чином були створені всі необхідні компоненти, а саме:

- `LoginComponent` – компонент, для реалізації сторінки для автентифікації користувача на стороні клієнта.
- `AdminComponent` – компонент, для адміністратора системи, на ньому він зможе бачити дані, що знаходяться в базі даних, а також цей компонент надаватиме інтерфейс для виклику функціоналу для заповнення бази даних.
- `HomeComponent` – компонент, що доступний всім користувачам системи, на ньому буде представлено карту-діаграму з країнами Європи на які можна буде натиснути для більш детальну інформацію про кожну країну, також на цьому компоненті буде знаходитись таблиця з інформацією про кожну країну. В цьому компоненті є інтерфейс для надання системі даних для обробки та на основі них робити прогнози на наступний рік.

- CountryComponent – компонент, що являє собою карту певної країни з діаграмою про демографічні показники, також в цьому компоненті будуть вкладки з різними роками (максимум 5 років), натискаючи на які користувач може дізнатись демографічну ситуацію минулих років. Також в цьому компоненті є інтерфейс для надання системі даних для обробки та на основі них робити прогнози на наступний рік.
- RegionComponent, CityComponent – компоненти системи, які за своїм функціоналом дуже схожі на компонент CountryComponent, вони надають схожий інтерфейс для користувача і можуть робити те саме, що і компонент вище. Було обрано розбити систему саме так, оскільки в майбутньому ці компоненти можуть відрізнитись і також в цьому компоненті є інтерфейс для надання системі даних для обробки та на основі них робити прогнози на наступний рік тільки вже для міст чи області (будь-якої найбільшої адміністративної одиниці країни).

Таким чином структура проекту має наступний вигляд (рис. 3.8):

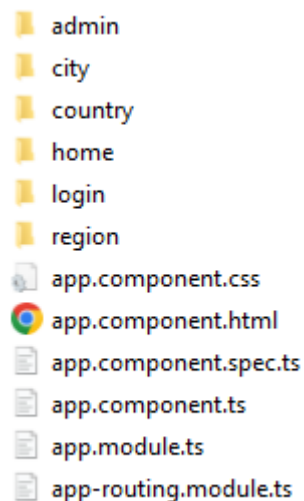


Рисунок 3.8 – Структура проекту зі всіма необхідними компонентами

Наступним кроком при створенні додатку було завантаження всіх необхідних бібліотек. Для використання бібліотеки PrimeNG її було імпортовано за допомогою команди `npm` (рис. 3.9).



```
C:\Users\TVG\Desktop\masters-work>npm install primeng
added 9 packages, and audited 10 packages in 8s
found 0 vulnerabilities
```

Рисунок 3.9 – завантаження бібліотеки PrimeNG до проекту

Для швидкої розробки потрібно використовувати перевагу, яку надає мова TypeScript перед JavaScript, а саме – типізація даних. Типізація даних дозволяє розробнику зрозуміти помилку ще на етапі створення продукту, наприклад, якщо в якомусь об'єкті буде відсутнє необхідне поле, то TypeScript не дозволить скомпілювати програму, поки помилка не буде вирішена на відміну від JavaScript (рис 3.10).

```
export interface Country {
  id: number;
  name: string;
  avgIncome: number;
  density: number;
  lifeExpectancy: number;
  population: number;
  educationRate: number;
  medicineRate: number;
}
```

Рисунок 3.10 – приклад використання інтерфейсів для типізації даних в мові TypeScript

Наведений вище приклад, дає зрозуміти, які поля отримає розробник коли буде звертатись до API.

Фреймворк Angular використовує підхід сервісів для доступу до даних при виклику API. Для отримання необхідної інформації про країну, було створено сервіс CountryService за допомогою тієї ж команди ng generate. Також, сервіси були створені і для інших компонентів, що використовують інформацію з API. Таким чином, була розділена логіка по отриманню даних та їх представлені (рис. 3.11).

```

@Injectables({
  providedIn: 'root'
})
export class CountryService {

  constructor(private http: HttpClient) {}

  getAllCountries() {
    return this.http.get('localhost:8080/api/v1/countries')
      .toPromise()
      .then(res => <Country[]> res)
      .then(data => { return data; });
  }
}

```

Рисунок 3.11 – частина сервісу CountryService

CountryService використовує інший сервіс HttpClient, який надає функції для обробки запитів на сервер. Також цей сервіс містить одну з багатьох функцій getAllCountries, що дозволяють вибрати всі країни за запитом на сервер. Також в цьому сервісі містяться функції для відправки даних з клієнта на сервер для запуску лінійної та поліноміальної регресій. Подібний функціонал міститься також і в інших сервісах.

Для використання доданого сервісу в компоненті Angular має механізм ін'єкції залежностей. Цей механізм дозволяє створювати непрямі залежності між сервісами та компонентами, а також між іншими сервісами, як було наведено в прикладі вище. В HomeComponent використовується метод життєвого циклу компоненту Angular, який дозволяє запустити якусь дію під час рендерингу компонента, виходить, коли користувач заходить на веб-додаток, то відпрацьовує метод onInit та виконує код, який міститься в ньому (рис. 3.12).

За допомогою виклику сервісу CountryService та методу getAllCountries було отримано всі необхідні дані про країни. Є різні імплементації моделі Country. Вона може використовуватись не тільки для виводу інформації в таблицю, а також для заповнення діаграм на карті Європи.

```

import { Component, OnInit } from '@angular/core';
import { Country } from '../interfaces/country';
import { CountryService } from '../services/country.service';

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.css']
})
export class HomeComponent implements OnInit {

  private countries: Country[] = [];

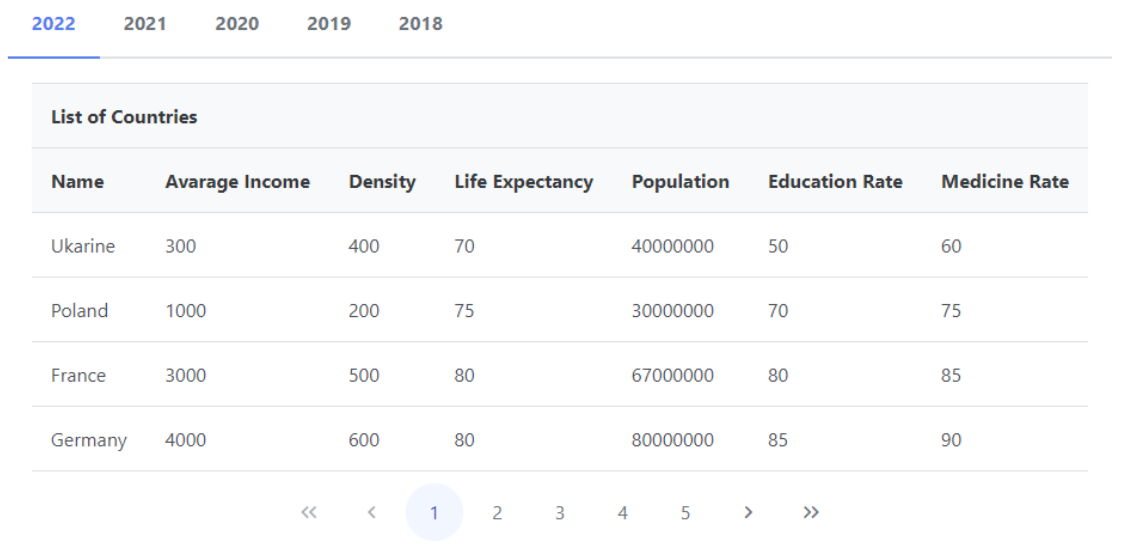
  constructor(private countryService: CountryService ) { }

  ngOnInit(): void {
    this.countryService.getAllCountries().then(countries => this.countries = countries);
  }
}

```

Рисунок 3.12 – приклад роботи сервісу

Коли користувач отримав всі необхідні дані, інформація виводиться йому за допомогою шаблону HTML. Angular використовує інтерполяцію для передачі інформації з компоненту на графічний інтерфейс користувача. Основними компонентами бібліотеки PrimeNG стали:



The screenshot shows a web interface with a navigation bar containing tabs for the years 2022, 2021, 2020, 2019, and 2018. Below the tabs is a table titled "List of Countries". The table has seven columns: Name, Average Income, Density, Life Expectancy, Population, Education Rate, and Medicine Rate. The data rows are for Ukraine, Poland, France, and Germany. Below the table is a pagination control showing a range of 5 pages, with the first page (1) selected.

| Name    | Average Income | Density | Life Expectancy | Population | Education Rate | Medicine Rate |
|---------|----------------|---------|-----------------|------------|----------------|---------------|
| Ukraine | 300            | 400     | 70              | 40000000   | 50             | 60            |
| Poland  | 1000           | 200     | 75              | 30000000   | 70             | 75            |
| France  | 3000           | 500     | 80              | 67000000   | 80             | 85            |
| Germany | 4000           | 600     | 80              | 80000000   | 85             | 90            |

Рисунок 3.13 – Приклад графічного інтерфейсу додатку з використанням таблиці, пагінації та вкладок з роками

- Table – для відображення інформації в таблиці.
- Button – для надання інтерфейсу взаємодії користувача з системою за допомогою натискання кнопки.
- TabView – для розбиття сторінки на 5 років, переключатись між якими можна натискаючи на цей компонент.

- Dialog – діалогове вікно для надання інформації від користувача і запуску регресій.

Було використано багато інших компонентів, які дозволили швидко побудувати інтерфейс користувача веб-додатку (рис. 3.13).

Наступним етапом в реалізації клієнтської частини додатку стало імпортування бібліотеки Highcharts, для виводу інформації про країни у вигляді графіків. Таким чином ця бібліотека дозволить спростити вивід інформації про показники країни і візуально побачити їх в тому чи іншому регіоні, місті або в Європі загалом.

Для використання бібліотеки було використано Highcharts Angular – це офіційна обгортка для Angular (рис. 3.14).

```
C:\Users\TVG\Desktop\lt\pop-client>npm install highcharts-angular --save
added 2 packages, removed 1 package, and audited 915 packages in 8s
121 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

Рисунок 3.14 – завантаження бібліотеки в проект

Також було імпортовано звичайну бібліотеку Highcharts, яка вже містить необхідні компоненти.

Дані про країну, чи регіон, чи місто можна передати в ці графіки і відображати їх на графічному інтерфейсі користувача. В кожному компоненті були додані різні карти.

В HomeComponent була додана вся карта Європи з доступними країнами, щоб користувач зміг на одну з них натиснути та перейти безпосередньо на сторінку з країною (рис. 3.15).

В CountryComponent була додана карта країни з регіонами на яких відображається інформація в залежності від того який фільтр вибрав користувач (рис. 3.16).

Для RegionComponent було зроблено теж саме, тільки тут можна бачити вже конкретний регіон країни з містами в ньому, аналогічно і для CityComponent.

## Country Selection

Select country for more information



Рисунок 3.15 – графічний інтерфейс користувача з картою усієї Європи та доступними країнами для вибору

Income by Region

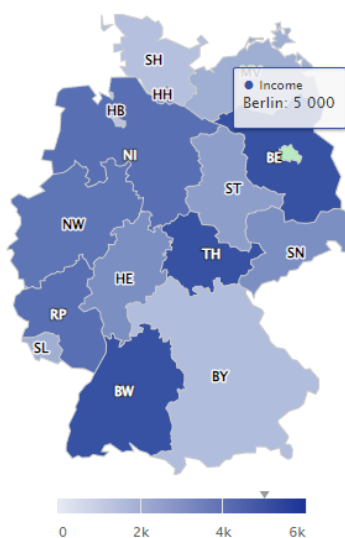


Рисунок 3.16 – Карта країни з вибором фільтру на середній дохід населення (average income)

Також, бібліотека Highcharts дозволяє нам вивести результати роботи лінійної та поліноміальної регресії (рис. 3.17).

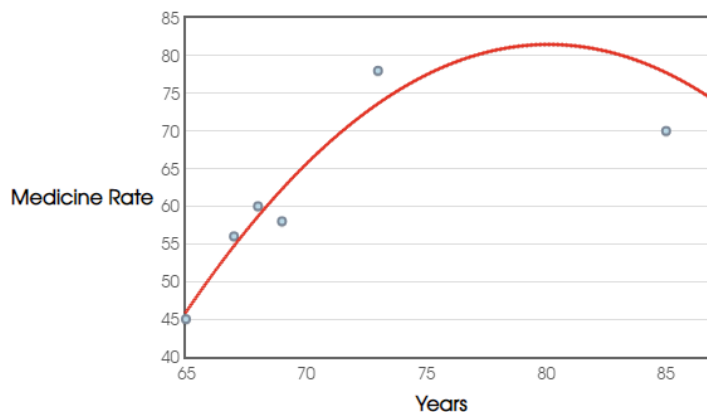


Рисунок 3.17 – результат роботи поліноміальної регресії у якої задано поліном другого ступеня

З роботи поліноміальної регресії можемо зробити висновок, що чим більший медичинський рейтинг, тим населення живе довше, якщо наприклад в нас рейтинг буде дорівнювати 80 то населення буде мати середню тривалість життя 80 років.

### 3.3 Реалізація серверної частини

Для написання API на мові C# було створено проект в Visual Studio за шаблоном web API (рис. 3.18).

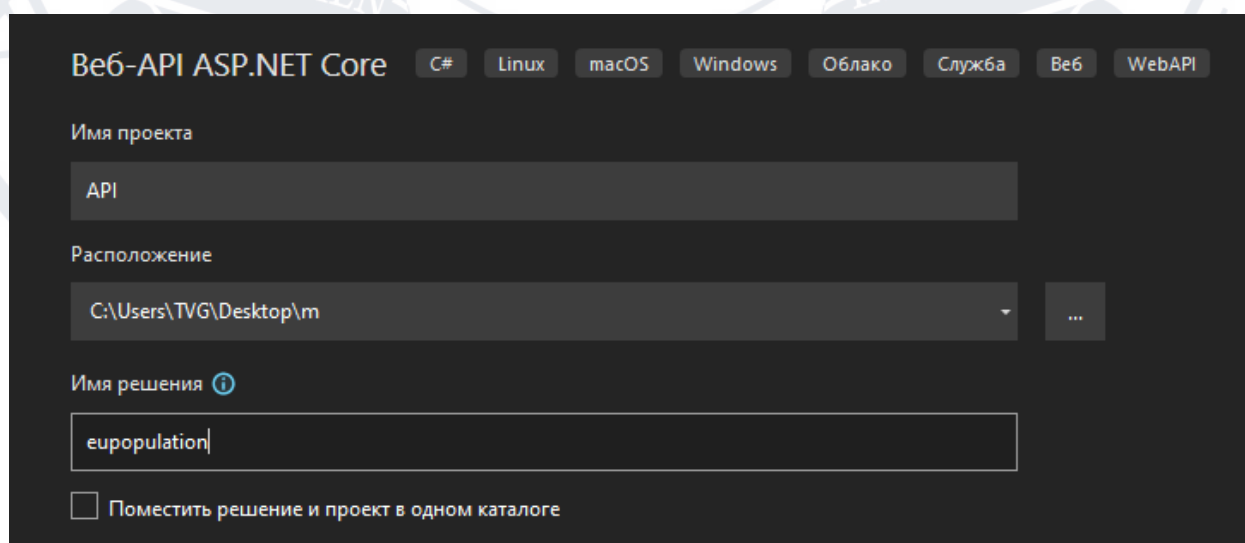


Рисунок 3.18 – Створення проекту за шаблоном веб-API

Рішення було поділено на 3 проекти:

- API – проект для контролерів, за допомогою якого сервер зможе відправляти інформацію користувачеві або отримувати її.
- Domain – проект для реалізації функціоналу взаємодії з базою даних, в ньому містяться моделі які описують сутності з бази даних, класи-репозиторії для доступу до даних, а також, клас контексту у якому міститься підключення до бази даних.
- Services – проект для реалізації шару бізнес логіки програми.

Загальна структура проекту виглядає наступним чином:

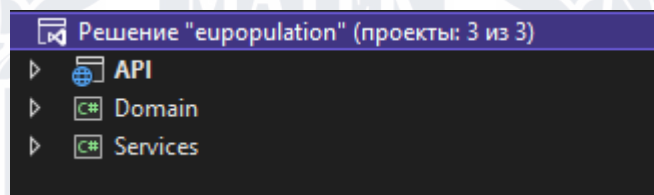


Рисунок 3.19 – Загальна структура проекту

В проекті Domain було реалізовано всі класи-моделі, які описують структуру даних в базі даних, а саме: City, Region, Country, SexPercentage, EducationPercentage, NationalityPercentage, CityYear, RegionYear, CountryYear, Year. Поля класу в моделях відображають колонки в відповідних таблицях в базі даних, а сам клас виступає у ролі таблиці.

Було додано клас контексту для підключення до бази даних, для цього був встановлений Entity Framework Core через менеджер пакетів NuGet. Клас контексту містить конструктор, а сам клас успадковується від класу DbContext (рис 3.20).

Щоб використовувати контекст даних в репозиторіях контекст було додано в сервіси за допомогою механізму ін'єкції залежностей. ASP.NET підтримує механізм ін'єкції залежностей за замовчуванням, що дозволяє легко додати даний контекст і використовувати його будь-де.

Було створено декілька репозиторії для взаємодії з даними, репозиторії являють собою класи в яких прописані методи, як потрібно взаємодіяти з даними, на окрему сутність в базі даних було створено окремий клас-репозиторій.

```

Ссылка: 1
internal class ApplicationContext : DbContext
{
    Ссылка: 0
    public DbSet<Country> countries { get; set; }
    Ссылка: 0
    public DbSet<Region> regions { get; set; }
    Ссылка: 0
    public DbSet<City> cities { get; set; }
    Ссылка: 0
    public DbSet<Year> year { get; set; }
    Ссылка: 0
    public DbSet<CityYear> cityYears { get; set; }
    Ссылка: 0
    public DbSet<RegionYear> regionYears { get; set; }
    Ссылка: 0
    public DbSet<CountryYear> countryYears { get; set; }
    Ссылка: 0
    public DbSet<SexPercentage> sexPercentages { get; set; }
    Ссылка: 0
    public DbSet<EducationPercentage> educationPercentages { get; set; }
    Ссылка: 0
    public DbSet<NationalityPercentage> nationalityPercentages { get; set; }

    Ссылка: 0
    public ApplicationContext()
    {
        Database.EnsureCreated();
    }
    Ссылка: 0
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseNpgsql("Host=localhost;Port=5433;Database=usersdb;Username=postgres;Password=1234");
    }
}

```

Рисунок 3.20 – клас контексту

Для реалізації бізнес логіки програми було створено декілька сервісів, а саме:

- DataService – сервіс, що відповідає за надання даних користувачеві, він зможе віддавати дані по роках, які в свою чергу будуть поділятися на дані по країнах, по областях, по містам. Отже клієнт зможе вибрати будь який рік і конкретну локацію яка його цікавить.
- ExportService – сервіс, що дозволяє експортувати дані в формат CSV, XLS, та інші формати, що підтримуються програмою Excel. Таким чином користувач зможе імпортувати дані на свій пристрій і використовувати їх за власним бажанням.

Для імпортування даних з бази даних в таблиці Excel було використано бібліотеку IronXL, яка була встановлена через пакетний менеджер NuGet.

Наступним етапом було створення шару контролерів, за допомогою механізму ін'єкції залежностей було надано доступ контролерам до сервісів.

Було створено два контролера:



- `DataController` – містить ендпойти для видачі інформації користувачеві про країну, місто чи регіон. Контролер може видавати інформацію як для представлення в таблиці, так і для представлення на карті.
- `ExportController` – містить один ендпоинт, для генерування файлу та надсилання його на клієнтську частину.

Щоб реалізувати API з використанням мови програмування Python було створено новий проект з використанням Django. Було виконано команди `pip install django` та `pip install django-rest-framework`.

Було також синхронізовано базу даних, яка була створена за допомогою моделей в минулому проекті. Та організована схожа структура. Були також створені сервіси:

- `AdminAuth` – сервіс, що відповідає за автентифікацію адміністратора, перш ніж адміністратор зможе використовувати можливості системи. Сервіс при автентифікації видає користувачеві токен, який в подальшому буде використовуватись для запитів на сервер.
- `AdminService` – сервіс, що надає функціонал для адміністратора (заповнення бази даних даними про демографію). Для доступу до сервісу має використовуватись токен, який було згенеровано в `AdminAuth` сервісі, це робиться для того, щоб звичайний користувач не міг додавати дані в базу даних системи.
- `ParseService` – сервіс для запуску веб-парсеру який зможе збирати дані з різних сайтів та заповнювати ними бази даних, це один з трьох способів заповнення даними таблиць.
- `ImportService` – сервіс, що дозволяє імпортувати дані з файлу, який завантажив адміністратор системи. Файл має обов'язково бути правильного формату та містити всі необхідні колонки.
- `AnalyzerService` – сервіс, який надає функціонал для аналізу даних про демографію. В ньому знаходяться функції для використання лінійної та мультилінійної регресії, а також кластеризації даних.

Серед всіх сервісів основним є сервіс `AnalyzerService` в якому міститься функціонал для виконання операцій лінійної та поліноміальних регресій (рис. 3.21).

```
def linear_regression(x, y, prediction):
    model = LinearRegression()
    model.fit(x, y)
    y_prediction = model.predict(prediction)
    return y_prediction

def polynomial_regression(x, y, predict, degree):
    transformer = PolynomialFeatures(degree=degree, include_bias=False)
    transformer.fit(x)
    x_ = transformer.transform(x)
    model = LinearRegression().fit(x_, y)
    y_pred = model.predict(predict)
    return y_pred
```

Рисунок 3.21 – функції для лінійної та поліноміальної регресії

У першому випадку ми маємо  $X$  та  $Y$  що є масивом даних, а `predict` це масив даних який ми хочемо передбачити. Для реалізації поліноміальної регресії було застосовано `PolynomialFeatures` які дозволяють побудувати криву та використати її для прогнозу, параметр `degree` вказує на те скільки вигинів буде мати крива.

Оцінки коефіцієнтів для методу звичайних найменших квадратів покладаються на незалежність ознак. Коли ознаки корельовані, а стовпці матриці планування мають приблизно лінійну залежність, матриця планування стає близькою до сингулярної, і в результаті оцінка методом найменших квадратів стає дуже чутливою до випадкових помилок у спостережуваній цілі, створюючи велику дисперсію. Така ситуація мультиколінеарності може виникнути, наприклад, коли дані збираються без плану експерименту.

Оскільки ці класи мають вже весь функціонал алгоритму лінійної регресії користувач може використовувати їх і не писати алгоритм методу найменших квадратів.

### 3.4 Умови застосування додатку

Даний проект був розроблений на основі ОС Windows, так як дана операційна система підтримує PostgreSQL сервер та мови програмування Python, C#, TypeScript. Однак, проект може базуватись на іншій системі, наприклад будь-який дистрибутив Linux, що може підтримувати перелічений вище функціонал.

Оскільки, даний продукт являє собою веб-додаток, все що потрібно користувачеві для користування даною програмою це веб-браузер та доступ до мережі інтернет.

Сам веб-додаток, необхідно розмістити на комп'ютерному сервері (хост), та дати йому домене ім'я, щоб його індексували пошукові системи та була змога знайти користувачеві даний програмний продукт в мережі.

Інсталяції додаткового програмного забезпечення для користування додаток не потребує.

#### **Висновки до розділу**

В даному розділі було розглянуто засоби реалізації додатку, спроектовану саму систему, детально описано процес розробки як клієнтської частини, так і серверної. Було продемонстровано основні моменти розробки додатку, а також наведені приклади графічного інтерфейсу користувача та роботи додатку в цілому.

## РОЗДІЛ 4

### АНАЛІЗ ДАНИХ ЗА ДОПОМОГОЮ ДОДАТКУ

#### 4.1 Графічний аналіз даних

Для тестування додатку було обрано Україну, та її демографічні показники з 2010 року. При зборі даних використовувалось декілька джерел, а саме:

- Сайт Організації Об'єднаних Націй, а саме, секція, що містить дані про демографію - <https://population.un.org/wpp/>.
- Сайт міністерства фінансів України, а саме, секція з середньою заробітною платою - <https://index.minfin.com.ua/ua/labour/salary/average/2010/>.
- Сайт Numbeo, на якому є інформація, щодо індексу охорони здоров'я по кожній з країн - <https://www.numbeo.com/health-care/>.

Було взято наступні демографічні показники:

- Загальне населення станом на 1 січня кожного року;
- населення чоловічої/жіночої статі станом на 1 січня кожного року;
- густота населення, та зміна населення (різниця народжуваності та смертей), також середній вік;
- середня заробітна плата по країні;
- індекс охорони здоров'я.

Сайт міністерства фінансів України пропонує дані за всі місяці кожного року починаючи з 2010, для полегшення обчислень, дані за всі місяці були додані одне до одного та розділені на 12 (кількість місяців).

Дані демографічні показники є дуже важливими як для бізнесу так і для державних установ. Для прикладу, якщо знати яким чином населення зменшується, можна виробити стратегію по запобіганню зменшенню та старінню населення, коли в країні може лишитись багато людей похилого віку, які вже не здатні працювати.

Дані про Україну можна звести в наступну таблицю (табл. 4.1)

Таблиця 4.1.1 – Таблиця демографічних показників України

| Рік  | Популяція країни (тисячі) | Популяція чоловіків (тисячі) | Популяція жінок (тисячі) | Густина населення | Середній вік | Зміна населення | Середній дохід населення | Індекс охорони здоров'я | Очікувана тривалість життя |
|------|---------------------------|------------------------------|--------------------------|-------------------|--------------|-----------------|--------------------------|-------------------------|----------------------------|
| 2010 | 45 778                    | 21 064                       | 24 619                   | 78,9              | 38,3         | - 198           | 2 247                    | 40.32                   | 70,6                       |
| 2011 | 45 588                    | 20 999                       | 24 517                   | 78,6              | 38,5         | - 157           | 2 639                    | 40.12                   | 71,4                       |
| 2012 | 45 444                    | 20 966                       | 24 440                   | 78,4              | 38,6         | - 145           | 3032                     | 40.28                   | 71,6                       |
| 2013 | 45 368                    | 20 940                       | 24 368                   | 78,2              | 38,8         | - 156           | 3 274                    | 47.89                   | 71,9                       |
| 2014 | 45 246                    | 20 884                       | 24 265                   | 77,9              | 38,9         | - 130           | 3 470                    | 47.10                   | 72,6                       |
| 2015 | 45 050                    | 20 822                       | 24 161                   | 77,6              | 39,1         | - 113           | 4 207                    | 44.27                   | 73,5                       |
| 2016 | 44 915                    | 20 762                       | 24 072                   | 77,4              | 39,4         | - 109           | 5 187                    | 49.69                   | 74,3                       |
| 2017 | 44 752                    | 20 687                       | 23 970                   | 77,1              | 39,6         | - 139           | 7 105                    | 49.81                   | 74,7                       |
| 2018 | 44 562                    | 20 595                       | 23 852                   | 76,7              | 39,9         | - 198           | 8 867                    | 50.11                   | 74,4                       |
| 2019 | 44 331                    | 20 481                       | 23 730                   | 76,3              | 40,2         | - 233           | 10 503                   | 50.95                   | 74,5                       |
| 2020 | 44 091                    | 20 332                       | 23 578                   | 75,8              | 40,5         | - 360           | 11 597                   | 52.33                   | 72,6                       |
| 2021 | 43 729                    | 20 147                       | 23 384                   | 75,1              | 40,8         | - 395           | 14 018                   | 53.43                   | 71,6                       |

Тепер, коли є дані, можна завантажувати їх у веб додаток.

Почнемо з побудови графіків, а саме – графіку зміни населення (рис. 4.1)

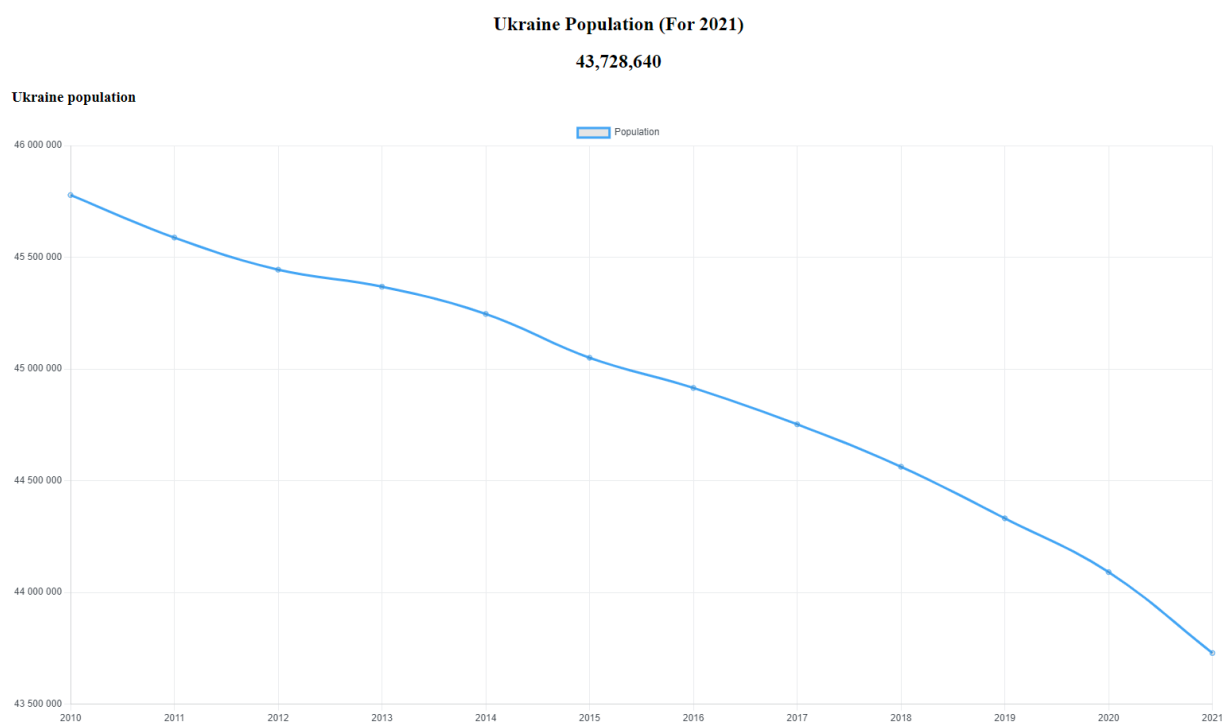


Рисунок 4.1 – Графік змінення населення України

Як можна бачити з кожним роком населення України невпинно зменшується. Тепер можна застосувати фільтр для відображення індекс охорони здоров'я на цьому ж графіку. Для коректного відображення на графіку, значення

населення було розділено на мільйон, для співставлення його до значення індексу охорони здоров'я (рис 4.2).

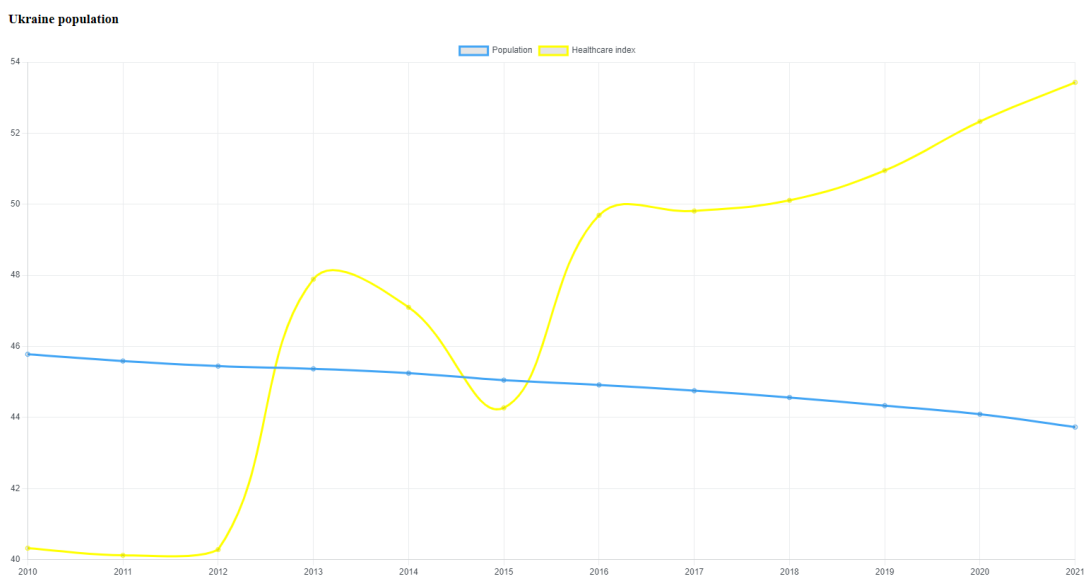


Рисунок 4.2 – Графік з показниками індексу охорони здоров'я та населення країни

З рисунка 4.2 можна зробити висновок, що індекс охорони здоров'я взагалі не впливає на зростання населення, більш того, можна бачити, що чим більше індекс охорони здоров'я, тим населення стає все меншим.

Побудуємо графік з параметрами індексу охорони здоров'я та середнього віку в країні та очікуваної тривалості життя (рис. 4.3).

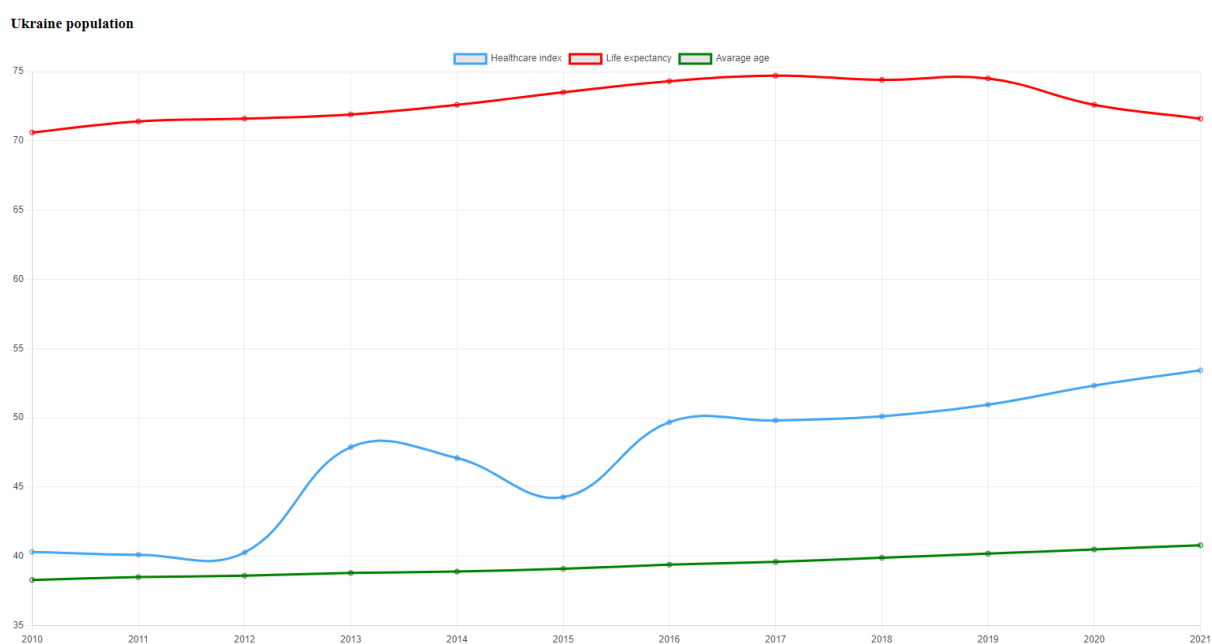


Рисунок 4.3 – Графік з показниками індексу охорони здоров'я очікуваної тривалості життя та середнього віку

З графіку можемо бачити, що середній вік в Україні збільшується, що означає старіння населення, графік зростає майже лінійно та не залежить від індексу охорони здоров'я. Очікувана тривалість життя постійно зростала, але з 2019 року маємо спад, в той час, коли індекс охорони здоров'я зростає в цей період.

#### 4.2 Прогнозування даних за допомогою лінійної та поліноміальної регресії

Тепер застосуємо лінійну регресію для передбачення показників очікуваної тривалості життя, середнього віку, та очікуваної тривалості життя, прогноз був зроблений на 3 роки. У якості значень, які будуть взяті для прогнозування були обрані: середній дохід, індекс охорони здоров'я, значення зміни населення. Для цих змінних були обрані значення, які відображають суму значення останнього року і середнє значення зміни цих показників, при умові, що тенденція зміни залишиться стабільною. Таблиця зі значеннями X для трьох наступних років має наступний вигляд (табл. 4.2):

Таблиця 4.2 – Таблиця зі значеннями X для прогнозування

| Рік  | Індекс охорони здоров'я | Середній дохід | Зміна населення |
|------|-------------------------|----------------|-----------------|
| 2022 | 54.43                   | 15 018         | -420            |
| 2023 | 55.43                   | 19 018         | -430            |
| 2024 | 55.43                   | 20 018         | -390            |

Після запуску лінійної регресії було отримано таблиці з наступними даними (рис. 4.4):

| Linear regression forecast for population |            |                  |            |                |            |                   |
|---|------------|------------------|------------|----------------|------------|-------------------|
| Year                                      | Population | Healthcare Index | Population | Average income | Population | Population change |
| 2022                                      | 44,060,647 | 54.43            | 43,558,402 | 15,018         | 43,757,182 | -420              |
| 2023                                      | 43,944,055 | 55.43            | 42,937,505 | 19,018         | 43,706,306 | -430              |
| 2024                                      | 43,827,462 | 56.43            | 42,782,281 | 20,018         | 43,909,807 | -390              |
|   |            | Healthcare Index |            | Average income |            | Population        |
|   | Coef       | -116592          |            | -155           |            | 5087              |
|   | Accuracy   | 0.79             |            | 0.96           |            | 0.56              |

Рисунок 4.4 – Таблиця з результатом роботи лінійної регресії для прогнозу кількості населення

Перша з трьох груп таблиць відображає дані по прогнозу чисельності населення в Україні протягом наступних трьох років, в ній є три колонки з показниками населення, а по праву сторону від кожної колонки знаходиться показники, за допомогою яких був зроблений прогноз. Нижня таблиця показує коефіцієнт лінійної регресії та точність обрахунку.

Як можна бачити найкращий результат лінійна регресія показала при прогнозуванні населення за допомогою середнього доходу цілих 96%, враховуючи, що прогноз відбувався за допомогою лінійної регресії такий відсоток точності є дуже суттєвим. Якщо вірити таблиці, у всіх трьох випадках населення буде скорочуватись, при обрахунку за допомогою середнього доходу видно, що населення скоротиться майже на мільйон в 2024 році. На другому місці по точності обрахунку є обрахунок за допомогою індексу охорони здоров'я – 79%, а найгіршим з усіх обрахунків виявився обрахунок за допомогою зміни населення – 56%. Можна зробити припущення, що середній дохід має великий вплив на населення, оскільки, при великій зарплаті люди зосереджені на будіванні кар'єри і заводять сім'ї пізніше, також це може пояснювати скорочення кількості багатодітних родин.

Наступним показником, що був прогнозований, це середній вік в країні (рис. 4.5).



| Lineral regression forecast for avarage age |             |                  |             |                |             |                   |
|---|-------------|------------------|-------------|----------------|-------------|-------------------|
| Year  | Avarage age | Healthcare Index | Avarage age | Avarage income | Avarage age | Population change |
| 2022  | 40.49       | 54.43            | 40.64       | 15,018         | 40.79       | -420              |
| 2023  | 41.14       | 55.43            | 41.95       | 19,018         | 42.15       | -430              |
| 2024  | 40.87       | 56.43            | 40.93       | 20,018         | 40.67       | -390              |
|   |             | Healthcare Index |             | Avarage income |             | Population        |
|   | Coef        | 0.15             |             | 0.00020188     |             | -0.00656123       |
|   | Accuracy    | 0.8              |             | 0.97           |             | 0.55              |

Рисунок 4.5 - Таблиця з результатом роботи лінійної регресії для прогнозу середнього віку в країні

З таблиці видно, що проноз на основі середнього доходу знову показав себе найкраще – 97% точності. Суттєвих змін в середньому віці населення не відбудеться, всі три варіанти показали, приблизно однаковий результат при зовсім різних рівнях точності, це свідчить про те, що середній вік населення доволі сталий показник, який змінюється не суттєво.

Остання таблиця яка була побудована, це таблиця, яка робила прогноз середньої тривалості життя (рис. 4.6).

| Lineral regression forecast for Life expectancy |             |                  |             |                |             |                   |
|---|-------------|------------------|-------------|----------------|-------------|-------------------|
| Year  | Avarage age | Healthcare Index | Avarage age | Avarage income | Avarage age | Population change |
| 2022  | 74          | 54.43            | 73.84       | 15,018         | 72.01       | -420              |
| 2023  | 74.16       | 55.43            | 74.31       | 19,018         | 71.97       | -430              |
| 2024  | 74.33       | 56.43            | 74.43       | 20,018         | 72.11       | -390              |
|   |             | Healthcare Index |             | Avarage income |             | Population        |
|   | Coef        | 0.16             |             | 0.0001189      |             | 0.00354665        |
|   | Accuracy    | 0.3              |             | 0.11           |             | 0.053             |

Рисунок 4.6 – Таблиця з результатом роботи лінійної регресії для прогнозу очікуваної тривалості життя

Як можна бачити, лінійна регресія в даному випадку показала найгірший результат – 30% в найкращому випадку, а найгірший випадок складає 5%

точності. Якщо вірити найкращому параметру для прогнозу, а саме індексу охорони здоров'я, то очікувана тривалість життя буде зростати, можна зробити висновок, що індекс охорони здоров'я відіграє важливу роль, при очікуваній тривалості життя, оскільки індекс охорони здоров'я відповідає в першу чергу за доступність послуг.

Як видно, лінійна регресія показує доволі таки велику точність при прогнозуванні одних демографічних даних за допомогою інших.

Для випадків, коли лінійна регресія виявилась не дуже точним методом прогнозу, можна застосувати поліноміальну регресію, для вирішення таких задач. Для цього було взято випадок, коли лінійна регресія показала результат у 5% точності, а саме – прогноз очікуваної тривалості життя на основі зміни населення.

Для того, щоб визначити скільки викривлень буде у кривій, пропонується побудувати графік, та на основі його задати це значення (рис.4.7).

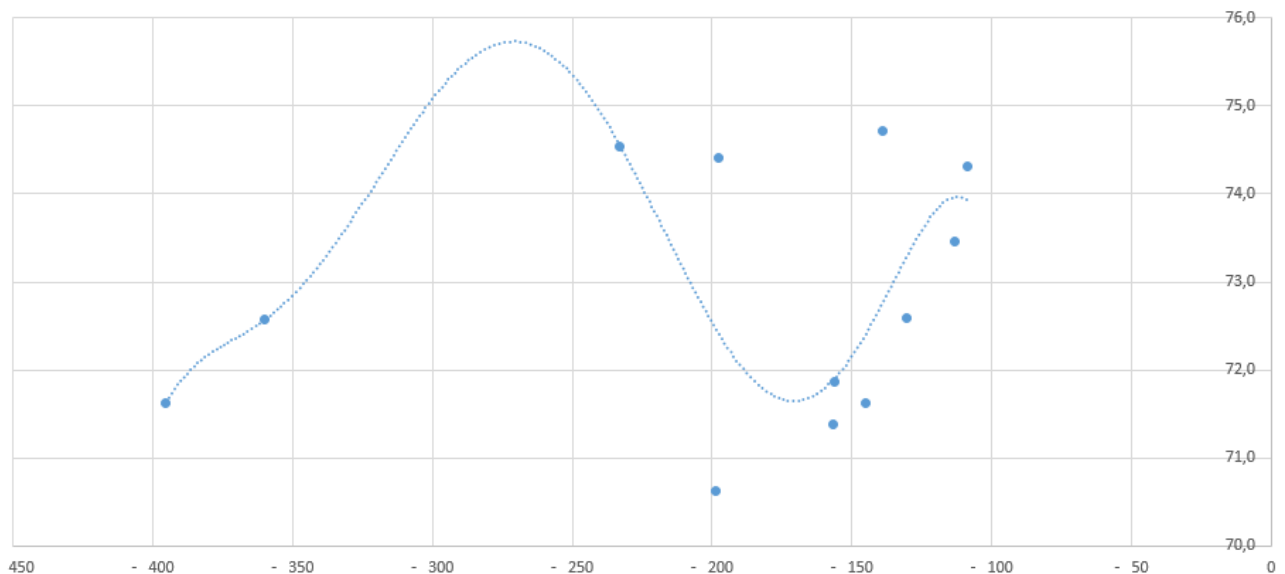


Рисунок 4.7 – Розподілення точок при залежності очікуваної тривалості життя від приросту населення

Як можна бачити з графіку застосовувати лінійну регресію тут не вигідно, через те, що будемо мати дуже велику різницю у відстані між лінією і кожною точкою.

Була побудована крива поліному 6-го степеню. При побудові такого графіку важливо, щоб крива не намагалась перетнути всі точки, оскільки це негативно впливає на точність.

Після обрахунків отримали наступні результати (рис. 4.8):

| Polynomial regression forecast for Life expectancy |                 |                   |
|--|-----------------|-------------------|
| Year   | Life expectancy | Population change |
| 2022   | 71.85           | -420              |
| 2023   | 71.32           | -430              |
| 2024   | 72.5            | -390              |
|  |                 | Population change |
| Coef   |                 | 0.15              |
| Accuracy   |                 | 0.35              |

Рисунок 4.8 – Застосування поліноміальної регресії

Було проведено декілька спроб, для виявлення найбільшої точності, і був підібраний поліном 6 ступеня, при використанні поліноміальної регресії можемо бачити, що значення точності змінилось з 5% на 30% це є дуже суттєво. Також бачимо, що очікувана тривалість життя дуже сильно змінюється, коли змінюється показник зміни населення, так коли він стає суттєво меншим, то очікувана тривалість життя, також стає суттєво меншою, що очікувано.

#### 4.3 Переваги та недоліки додатку

Даний веб-додаток містить ряд переваг та недоліків у порівнянні з аналогами. Серед переваг можна відмітити можливість для користувача під час виконання операцій прогнозування вносити власні дані. Таким чином, якщо є дані на наступні декілька років серед будь яких показників, можна зробити прогноз на інші показники базуючись на даних які були введені користувачем. Також, до переваг додатку можна віднести можливість фільтрувати графіки, тобто показувати на діаграмі тільки один або декілька графіків. Це може знадобитись користувачеві, коли він слідкує за тим як два різних показників змінювались на графіку. Оскільки, у якості осі X виступає завжди рік, а Y це будь

яке інше числове значення, то такий підхід дозволяє будувати декілька графіків на одній і тій самій площині.

До головного недоліку додатку можна віднести малу кількість алгоритмів для прогнозу, наприклад, можна також застосовувати для цієї задачі алгоритм мультилінійної регресії, який дозволяє будувати прогноз на основі двох та більше змінних, що дозволяє підняти точність. Також, недоліком є вузька область (тільки демографія країн Європи) аналогів цього додатку зазвичай покривають населення всієї планети, хоча, більш вузька область дозволяє вивчити її більш поглиблено.

### **Висновки до розділу**

В даному розділі було розглянуто роботу програми. Було завантажено дані з декількох джерел про демографічну ситуацію України. Був проведений графічний аналіз даних, було встановлено декілька закономірностей щодо впливу різних чинників на зростання населення. Також, були застосовані методи лінійної та поліноміальної регресії для демографічних показників на наступні декілька років. Як було продемонстровано, для прогнозування даних лінійна регресія в даному випадку має доволі велику точність в середньому 75%. Якщо дані будуть мати більшу точність прогнозу зросте. Для випадків, де звичайна лінійна регресія продемонструвала низьку точність було застосовано поліноміальну регресію, що дозволило збільшити точність.

## ВИСНОВКИ

В даній роботі було створено веб-додаток за допомогою якого можна робити прогнози по зміні демографічної ситуації в країнах Європи.

Було досліджено різні інструменти розробки для побудови даного веб-додатку такі як: Python, C#, TypeScript, Angular, ASP.NET, Django REST Framework, також бібліотеки регресивного аналізу. Було спроектовано архітектуру даного додатку, базу даних, а також запроваджено реалізацію алгоритмів лінійної та поліноміальної регресії використовуючи готові рішення з бібліотеки Python.

При проектуванні додатку було обрано архітектурний підхід REST. Для створення серверної частини використовувалось такі мови програмування, як C# та Python, для створення клієнтської частини було використано фреймворк Angular та мову програмування TypeScript.

Було зроблено декілька тестів для перевірки роботи лінійної та поліноміальної регресії, виведені відсотки точності, прогнозовані результати, а також коефіцієнти. Дослідження показали, що навіть лінійна регресія показує гарний відсоток точності при прогнозуванні демографічних показників на наступні 3 роки. Коли точності не вистачало, можна було застосувати поліноміальну регресію, поліном підбирався таким чином, щоб регресія не втратила у відсотках точності, оскільки, при збільшені ступеня поліному можна зіштовхнутись з проблемою того, що крива почне «точно» перетинати кожную точку, що дасть менший відсоток точності обрахунків.

Тест робився на демографічних даних, які були взяті з офіційних джерел, таких як: Міністерство фінансів України, а також Організації Об'єднаних націй. Робився графічний аналіз, по графікам можна було бачити як одні показники впливають на інші, наприклад, якщо вони синхронно змінюються в один і той самий рік. За допомогою методів лінійної та поліноміальної регресій можна було встановити деякі залежності одних показників від інших.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. С. В. Григор'єва «Основи демографії» Текст лекцій, НТУ «ХП», Харків 2010.
2. Сучасна демографічна ситуація в Україні URL: <https://www.mediku.com.ua/suchasna-demografichna-situaciya-v-ukrayini/> (дата звернення: 11.05.2022).
3. Методологічні підходи і методи аналізу демографічних процесів і структур. URL: <https://lektsii.com/1-172857.html> (дата звернення: 11.05.2022).
4. Demographics: How to Collect, Analyze, and Use Demographic Data URL: <https://www.investopedia.com/terms/d/demographics.asp> (дата звернення: 16.05.2022).
5. Worldometer URL: <https://uk.wikipedia.org/wiki/Worldometer> (дата звернення: 16.05.2022).
6. World Population Review URL: <https://www.crunchbase.com/organization/world-population-review> (дата звернення: 16.05.2022).
7. Our World in Data URL: [https://en.wikipedia.org/wiki/Our\\_World\\_in\\_Data](https://en.wikipedia.org/wiki/Our_World_in_Data) (дата звернення: 16.05.2022).
8. REST API Architectural Constraints Review URL: <https://www.geeksforgeeks.org/rest-api-architectural-constraints/> (дата звернення: 14.09.2022).
9. REST Architectural Constraints URL: <https://restfulapi.net/rest-architectural-constraints/> (дата звернення: 17.09.2022).
10. Многоуровневая архитектура URL: <https://metanit.com/sharp/mvc5/23.5.php> (дата звернення: 17.09.2022).
11. What is a Framework in Programming & Why You Should Use One URL: <https://www.netsolutions.com/insights/what-is-a-framework-in-programming/> (дата звернення: 17.09.2022).
12. ML | Linear Regression URL: <https://www.geeksforgeeks.org/ml-linear-regression/> (дата звернення: 19.09.2022).
13. Why linear regression is important URL: <https://www.ibm.com/topics/linear-regression> (дата звернення: 19.09.2022).

14. Linear regression URL: [https://en.wikipedia.org/wiki/Linear\\_regression](https://en.wikipedia.org/wiki/Linear_regression) (дата звернення: 19.09.2022).
15. Поліноміальна регресія URL: <https://uk.wikipedia.org/wiki/> (дата звернення: 20.09.2022).
16. Python Programming Language URL: <https://www.geeksforgeeks.org/python-programming-language/> (дата звернення: 20.09.2022).
17. What Is Python Used For? A Beginner's Guide URL: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python> (дата звернення: 20.09.2022).
18. A tour of the C# language URL: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/> (дата звернення: 20.09.2022).
19. A-level Computing/AQA/CSharp URL: [https://en.wikibooks.org/wiki/A-level\\_Computing/AQA/CSharp](https://en.wikibooks.org/wiki/A-level_Computing/AQA/CSharp) (дата звернення: 20.09.2022).
20. Angular (фреймворк) URL: <https://uk.wikipedia.org/wiki/Angular> (дата звернення: 20.09.2022).
21. The Good and the Bad of Angular Development URL: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/> (дата звернення: 20.09.2022).
22. Введение в Angular URL: <https://metanit.com/web/angular2/1.1.php> (дата звернення: 21.09.2022).
23. What is ASP.NET? URL: <https://umbraco.com/knowledge-base/aspnet/> (дата звернення: 10.10.2022).
24. Django REST framework URL: <https://www.django-rest-framework.org/> (дата звернення: 11.10.2022).
25. How to create a REST API with Django REST framework URL: <https://blog.logrocket.com/django-rest-framework-create-api/> (дата звернення: 11.10.2022).

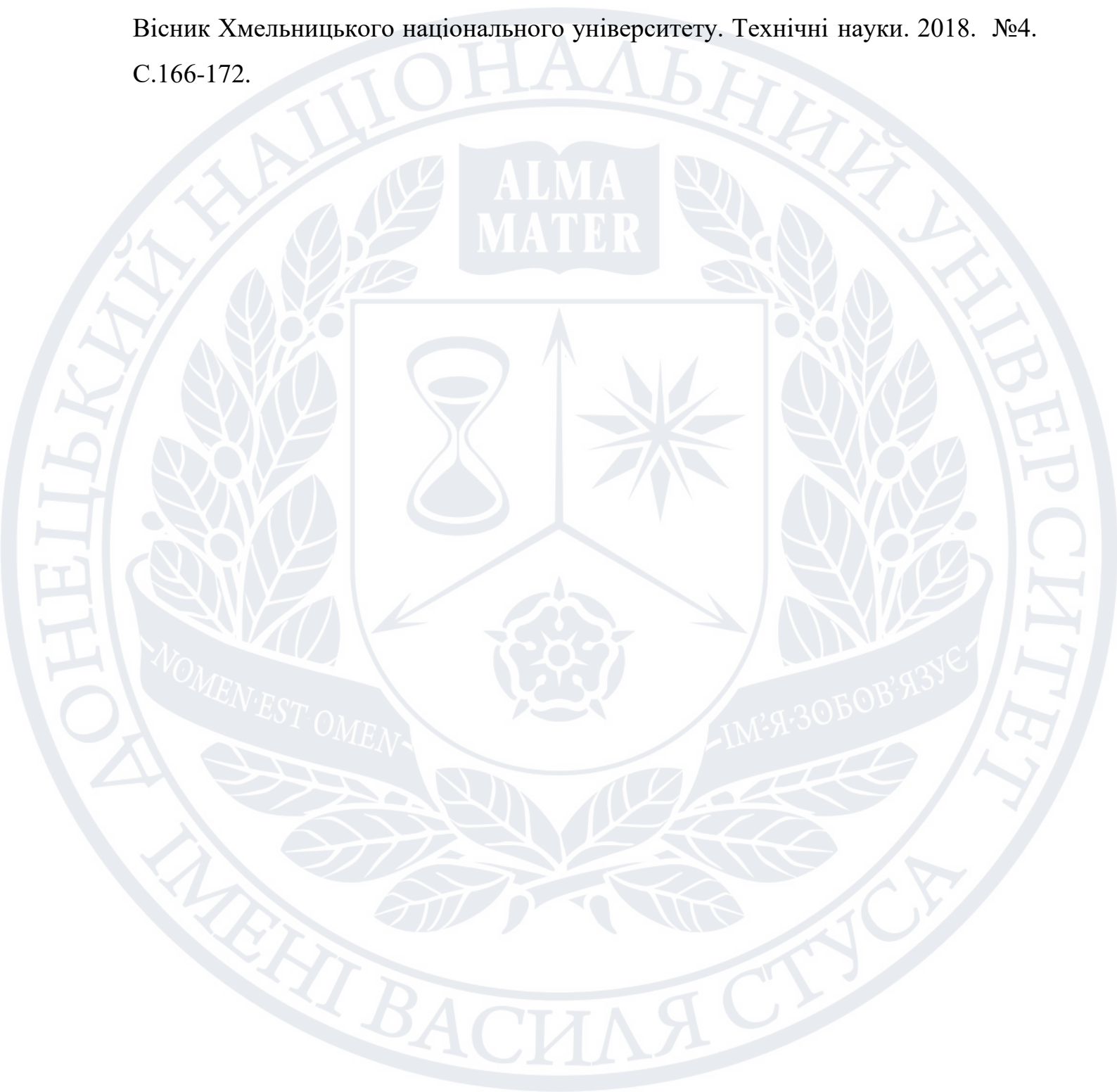
26. Beautiful Soup Documentation URL:  
<https://www.crummy.com/software/BeautifulSoup/bs4/doc/> (дата звернення:  
11.10.2022).
27. Entity Framework Core URL:  
<https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx> (дата  
звернення: 12.10.2022).
28. Angular PrimeNG Introduction URL:  
<https://www.geeksforgeeks.org/angular-primeng-introduction/> (дата звернення:  
12.10.2022).
29. Highcharts URL: <https://www.highcharts.com/> (дата звернення:  
12.10.2022).
30. Documentation for Visual Studio Code URL:  
<https://code.visualstudio.com/docs> (дата звернення: 12.10.2022).
31. Welcome to the Visual Studio IDE URL: [https://learn.microsoft.com/en-  
us/visualstudio/get-started/visual-studio-ide?view=vs-2022](https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022) (дата звернення:  
12.10.2022).
32. PyCharm URL: [https://www.jetbrains.com/help/pycharm/quick-start-  
guide.html](https://www.jetbrains.com/help/pycharm/quick-start-guide.html) (дата звернення: 12.10.2022).
33. Features Of PostgreSQL URL: [https://www.w3schools.blog/features-  
postgresql](https://www.w3schools.blog/features-postgresql) (дата звернення: 12.10.2022).
34. PostgreSQL: The world's most advanced open source database URL:  
<https://www.postgresql.org/> (дата звернення: 12.10.2022).
35. Demography | National Geographic Society URL:  
<https://education.nationalgeographic.org/resource/demography> (дата звернення:  
12.10.2022).
36. Demography – Wikipedia URL:  
<https://en.wikipedia.org/wiki/Demography> (дата звернення: 15.10.2022).
37. Demographic Analysis - an overview | ScienceDirect Topics URL:  
<https://www.sciencedirect.com/topics/social-sciences/demographic-analysis> (дата  
звернення: 15.10.2022).



38. Demographics & Lifestyle Analysis URL: <https://economicdevelopment.extension.wisc.edu/articles/demographics-lifestyle-analysis/> (дата звернення: 15.10.2022).
39. Documentation – PostgreSQL URL: <https://www.postgresql.org/docs/> (дата звернення: 15.10.2022).
40. ASP.NET documentation - Microsoft Learn URL: <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0> (дата звернення: 15.10.2022).
41. scikit-learn: machine learning in Python — scikit-learn 1.1.3 URL: <https://scikit-learn.org/stable/> (дата звернення: 15.10.2022).
42. Руководство по ASP.NET Core 7 – Metanit URL: <https://metanit.com/sharp/aspnet6/> (дата звернення: 15.10.2022).
43. Visual Studio documentation - Microsoft Learn URL: <https://learn.microsoft.com/en-us/visualstudio/windows/?view=vs-2022> (дата звернення: 15.10.2022).
44. Getting started with Angular URL: <https://angular.io/start> (дата звернення: 15.10.2022).
45. All you need to know about Polynomial Regression URL: <https://www.analyticsvidhya.com/blog/2021/07/all-you-need-to-know-about-polynomial-regression/> (дата звернення: 15.10.2022).
46. Python Machine Learning Polynomial Regression - W3Schools URL: [https://www.w3schools.com/python/python\\_ml\\_polynomial\\_regression.asp](https://www.w3schools.com/python/python_ml_polynomial_regression.asp) (дата звернення: 20.10.2022).
47. Machine Learning: Polynomial Regression with Python URL: <https://towardsdatascience.com/machine-learning-polynomial-regression-with-python-5328e4e8a386> (дата звернення: 20.10.2022).
48. Machine Learning Algorithms – Javatpoint URL: <https://www.javatpoint.com/machine-learning-algorithms> (дата звернення: 20.10.2022).

49. Django REST Swagger URL: <https://django-rest-swagger.readthedocs.io/en/latest/> (дата звернення: 20.10.2022).

50. Січко Т.В., Юрчук Б.О. Розробка веб-додатку туристичної фірми. Вісник Хмельницького національного університету. Технічні науки. 2018. №4. С.166-172.



Талаш Владислав Геннадійович  
Прізвище, ім'я по батькові

Інформаційних і прикладних технологій  
Факультет

122 Комп'ютерні науки  
Шифр і назва спеціальності

Комп'ютерні технології обробки даних (Data Science)  
Освітня програма

## **ДЕКЛАРАЦІЯ АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ**

Усвідомлюючи свою відповідальність за надання неправдивої інформації, стверджую, що подана кваліфікаційна (магістерська) робота на тему **«РОЗРОБКА ВЕБ-ДОДАТКУ АНАЛІЗУ ДЕМОГРАФІЧНИХ ПОКАЗНИКІВ КРАЇН ЄВРОПИ»**

є написаною мною особисто.

Одночасно заявляю, що ця робота:

- не передавалась іншим особам і подається до захисту вперше;
- не порушує авторських та суміжних прав, закріплених статтями 21-25 Закону України «Про авторське право та суміжні права»;
- не отримувалась іншими особами, а також дані та інформація не отримувалась у недозволеній спосіб.

Я усвідомлюю, що у разі порушення цього порядку моя кваліфікаційна робота буде відхилена без права її захисту, або під час захисту за неї буде поставлена оцінка «незадовільно».

---

(дата)

---

(підпис здобувача освіти)