

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ЯКУБИЧ КАТЕРИНА ОЛЕКСІЇВНА

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій
к. т. н., доцент
_____ О. В. Зелінська
« ____ » _____ 2024 р.

**ІНТЕЛЕКТУАЛЬНА ІНФОРМАЦІЙНО-АНАЛІТИЧНА
СИСТЕМА ДОНОРСТВА КРОВІ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (магістерська) робота

Науковий керівник:
О. В. Зелінська, доцент кафедри
інформаційних технологій,
к. т. н., доцент

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

АНОТАЦІЯ

Якубич К. О. Інтелектуальна інформаційно-аналітична система донорства крові. Спеціальність 122 «Комп'ютерні науки», Освітня програма «Комп'ютерні технології обробки даних (Data Science)». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

У кваліфікаційній (магістерській) роботі досліджено поняття інтелектуальної інформаційно-аналітичної системи для донорства крові. Була розроблена інтелектуальна інформаційно-аналітична система для донорства крові для міста Вінниці.

Ключові слова: інтелектуальна, інформаційна, аналітична, система, донорство крові, аналіз даних, JavaScript, ReactJS, MongoDB, Telegram API.

88 с., 32 рис., 52 джерело.

Yakubych K. O. Intellectual information and analytical system of blood donation. Specialty 122 "Computer science", Programme "Data Science". Vasyl' Stus Donetsk National University, Vinnytsia, 2024.

In the qualification (master's) thesis, the concept of an intelligent information and analytical system for blood donation was investigated. An intelligent information and analytical system for blood donation for the city of Vinnytsia was developed.

Keywords: intelligent, information, analytical, system, blood donation, data analysis, JavaScript, ReactJS, MongoDB, Telegram API.

Fig. 32. Bibliography: 52 items.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	4
ВСТУП	5
РОЗДІЛ 1 ПОСТАНОВКА ЗАДАЧІ І ОГЛЯД АНАЛОГІВ ІСНУЮЧИХ ДОДАТКІВ.....	8
1.1 Опис актуальності інтелектуальної інформаційно-аналітичної системи донорства крові.....	8
1.2 Дослідження існуючих систем донорства крові	11
1.3 Постановка задачі.....	14
Висновок до розділу 1	16
РОЗДІЛ 2 ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ ПРОГРАМНОГО ПРОДУКТУ	17
2.1 Вибір методів збору та аналізу даних	17
2.2 Вибір інструментів для створення інтелектуальної інформаційно- аналітичної системи	22
2.3 Аналіз та вибір мов програмування для розробки	34
2.4 Створення моделі інтелектуальної інформаційно-аналітичної системи... 47	47
Висновок до розділу 2	52
РОЗДІЛ 3 РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНО- АНАЛІТИЧНОЇ СИСТЕМИ ДОНОРСТВА КРОВІ	53
3.1 Аналіз вимог до системи та її функціональні можливості.....	53
3.2 Створення бази даних та її структура.....	55
3.3 Програмна реалізація.....	64
3.4 Аналіз отриманих результатів.....	69
Висновок до розділу 3	80
ВИСНОВКИ.....	81
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	83

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

ПАС – інтелектуальна інформаційно-аналітична система

AI – artificial intelligence (штучний інтелект)

ML – machine learning (машинне навчання)

JS – JavaScript (мова програмування)

API – application programming interface (програмний інтерфейс)

IDE – integrated development environment (інтегрована середовище розробки)

GUI – graphical user interface (графічний інтерфейс)

ВСТУП

Сучасні технології та інформаційні системи знаходять все більше застосувань у сфері охорони здоров'я та медичних послуг. Одним із важливих напрямків в цій галузі є донорство крові, яке є життєво важливим елементом медичної допомоги та рятувницької практики. Забезпечення стабільних запасів крові та ефективного управління донорами вимагають вдосконалення традиційних підходів та використання новітніх рішень.

У цьому контексті інтелектуальна інформаційно – аналітична система донорства крові виступає як інноваційний інструмент, спрямований на покращення ефективності управління процесом донорства крові. Ця система базується на використанні аналітичних алгоритмів та передових технологій збору та обробки даних.

Метою роботи є дослідження та розробка інтелектуальної інформаційно – аналітичної системи донорства крові. Робота спрямована на розкриття функціональних можливостей системи, вивчення її переваг та потенціалу для покращення процесу донорства крові, а також аналіз даних та виявлення факторів, що впливають на ефективність донорства.

Об'єктом дослідження є інформаційно-аналітична система донорства крові.

Предметом дослідження є методи для створення інтелектуальної інформаційно-аналітичної системи донорства крові для міста Вінниці.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Огляд та аналіз аналогів інтелектуальних інформаційно-аналітичних систем донорства крові. Висвітлення їх переваг та недоліків.
2. Дослідження методів для збору та аналізу даних, а також основних складових системи донорства крові, включаючи реєстрацію донорів, управління запасами крові, організацію донорських сеансів, сповіщення та нагадування донорам.

3. Побудова моделі інформаційно – аналітичної системи донорства крові.
4. Вибір інструментів для реалізації інтелектуальної інформаційно – аналітичної системи донорства крові.
5. Реалізація інтелектуальної інформаційно – аналітичної системи донорства крові для міста Вінниця.
6. Формування бази даних донорів і їх аналіз для оцінки ситуації та прогнозування.

Методи збору та аналізу даних які використовуються:

1. Реєстрація донорів: система збирає основні персональні дані потенційних донорів.
2. Медична інформація: збір інформації про стан здоров'я донора, тимчасові протипоказання.
3. Інформація про потребу в крові: система має дані про потребу в крові на станції переливання крові.
4. Кластерний аналіз: групування донорів на основі схожих характеристик.
5. Аналіз асоціацій: виявляються зв'язки між різними характеристиками (наприклад, вік, група крові) та донорською активністю.
6. Прогнозування запитів: прогнозується кількість доступних донорів за певний період.

Наукова новизна полягає у впровадженні інтелектуальної інформаційно-аналітичної системи донорства крові для міста Вінниця.

Результати даного дослідження мають важливе практичне значення для медичних установ, банків крові та організацій, що займаються донорством крові. Впровадження інтелектуальної інформаційно – аналітичної системи донорства крові може покращити процес залучення та управління донорами, забезпечити стабільність запасів крові, зменшити витрати та час на організацію донорських сеансів, а також покращити аналітичні можливості для планування та прийняття рішень.

Завдяки цьому дослідженню буде отримана глибша інформація про роль та потенціал інтелектуальної інформаційно – аналітичної системи донорства крові у сучасній медицині. Його результати сприятимуть подальшій розробці та вдосконаленню інформаційних систем в галузі донорства крові, сприяючи збереженню життя та покращенню якості медичної допомоги.

Результати дослідження були висвітлені у таких наукових публікаціях:

- Якубич К.О., Зелінська О.В. Інформаційно-аналітична система донорства крові. Матеріали IV всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених. Вінниця. 2023. С. 202 – 204.
- Зелінська О.В., Гавліцький В.Ф., Якубич К.О. Проектування інтелектуальної інформаційно-аналітичної системи донорства крові. Електронний журнал «Наука і техніка сьогодні». Випуск №13(27). Київ. 2023. С. 748 – 758.

Кваліфікаційна (магістерська) робота складається із вступу, трьох розділів із висновками, загального висновку, списку використаних джерел.

Загальний обсяг кваліфікаційної роботи становить 88 сторінки. Обсяг основної частини роботи складає 81 сторінок.

РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ І ОГЛЯД АНАЛОГІВ ІСНУЮЧИХ ДОДАТКІВ

1.1 Опис актуальності інтелектуальної інформаційно-аналітичної системи донорства крові

В сучасному світі проблеми, пов'язані зі забезпеченням достатніх запасів безпечної крові, залишаються нагальними і викликають серйозні виклики для медичної системи. Недостатність крові може становити загрозу життю пацієнтів, що потребують трансфузії, і погіршувати результати лікування. Оптимізація процесу донорства крові та ефективне управління донорами стають важливими завданнями для забезпечення безпеки та якості медичних послуг.

Україна завжди стикалася із дефіцитом донорської крові. У країні існують 47 спеціалізованих установ та переливальних станцій, таких як центри крові, які забезпечують 56% загального обсягу заготовленої крові та її компонентів. Додатково, 44% цього обсягу управляють 327 спеціалізованими відділеннями закладів охорони здоров'я, включаючи відділення переливання крові та трансфузіології в 83 лікарнях, де часто відсутні відповідні умови і обладнання для стандартизації технологічних процесів і контролю їхнього виконання. Система працює неспрямовано, відсутня чітка координація, структура, підпорядкування та чіткий розподіл повноважень, і також відсутня ефективна міжгалузева взаємодія.

Сутність проблеми полягає в тому, що в Україні донорська активність є спонтанною або обумовленою обставинами. Експерти приходять до висновку, що українці виявляють бажання здавати кров лише у випадках, коли хтось з їхніх близьких потрапляє до лікарні або внаслідок надзвичайних подій, таких як воєнний конфлікт. У країні відсутня культура регулярного донорства, коли кожна здорова людина розуміє важливість систематичного внеску крові.

Україна має 437,425 зареєстрованих донорів крові та її компонентів, що складає приблизно 1,025% від загальної кількості населення. Ці донори

здійснили 623,920 донацій, що означає, що на кожного донора припадає приблизно 1,43 донації щорічно.

У контексті якісного складу донорського руху в Україні, лише 10,40% або 45,511 осіб, є активними, з 437,425 зареєстрованих донорів. Активні донори визначаються як ті, хто регулярно, три і більше разів щорічно, здає кров та її компоненти, при цьому не перевищуючи 5 разів на рік. Ці особи є стабільним основним елементом донорського руху, вони, як правило, здають кров безоплатно та регулярно піддаються медичним оглядам. Стан їхнього здоров'я піддається контролю, зокрема за індикаторами інфекційних захворювань.

На жаль, у більшості випадків, а саме 89,6%, донори є первинними. Це означає, що вони здають кров вперше або роблять це як донори резерву, тобто не регулярно і не більше двох разів на рік. Серед них можна виділити дві категорії: ті, хто здає кров для отримання грошової компенсації, та донори-родичі, які здають кров для лікування своїх близьких. Це призводить до ситуативних дефіцитів та нестабільності в обсязі кроводач.

Важливо регулярно поповнювати запаси крові, особливо зараз, в період війни. У перші місяці повномасштабного конфлікту було помітно зростання числа людей, які активно відвідували Центри крові для донорства. Проте цей підйом тривав лише два-три місяці, і навіть хоча на той момент було надлишок компонентів донорської крові, з часом активність зменшилася. З метою забезпечення стабільних запасів було створено стратегічний запас та реєстр донорів. Але з плином часу люди поступово адаптувалися до умов війни, і загальна активність здавання крові зменшилася.

У період війни, донорська кров стає важливим і, можливо, єдиним засобом рятувального втручання для поранених осіб, які стикаються з важкими травмами та крововтратою. Головним завданням тих, хто знаходиться в безпечному тилу, є використання всіх можливих зусиль для збереження життя тих, хто постраждав через російське вторгнення. Особливо важливим стає збільшення обсягу донорської крові, оскільки чим більше людей, зокрема військових, мають спільну групу крові, тим більше ця кров може бути використана. Забезпечення

стабільних запасів крові стає ключовим завданням, оскільки кров має обмежений термін придатності в залежності від її компонентів: цільна кров – 30 днів, тромбоцити – до 5 днів, свіжа плазма крові – не більше 3 днів.

Окрім людського впливу, існують технічні виклики, які ускладнюють ефективне управління цим процесом. Багато систем донорства крові використовують застарілі методи управління та обліку донорів, що може призводити до неконтрольованих ризиків і недостачі крові.

Інтелектуальна інформаційно-аналітична система донорства крові виступає як потужний інструмент, що може вирішити багато складнощів і поліпшити ефективність системи донорства крові. Однією з головних переваг інтелектуальної інформаційно – аналітичної системи є можливість автоматизації багатьох процесів, пов'язаних з донорством крові, що сприяє зниженню людського фактору та помилок, а також прискорює обробку і аналіз даних.

Завдяки використанню штучного інтелекту, аналітичних алгоритмів та передових технологій, інтелектуальна інформаційно – аналітична система може забезпечити точну і швидку інформацію про наявність донорів, їх медичні дані, а також управляти запасами крові та планувати донорські сеанси. Вона дозволяє виявляти нові тенденції, аналізувати дані та прогнозувати потреби в крові, що є важливим для забезпечення постійної наявності достатнього обсягу крові та попередження дефіциту.

Окрім того, інтелектуальна інформаційно – аналітична система донорства крові може покращити комунікацію та взаємодію між медичними установами, банками крові, донорами та отримувачами крові. Вона надає можливість сповіщати донорів про можливість здати кров, нагадувати їм про наступні донорські сеанси та забезпечувати інформаційну підтримку щодо вимог та процедур. Це сприяє залученню більшої кількості донорів та покращує їх здатність доцільно планувати свої донорські візити.

Важливим аспектом системи є забезпечення конфіденційності та безпеки даних. Всі медичні та персональні дані донорів повинні бути захищені від несанкціонованого доступу та використання. Інтеграція відповідних механізмів

безпеки та шифрування є ключовими факторами для успішного впровадження системи донорства крові.

Однак, необхідно також враховувати потенційні виклики та обмеження, пов'язані з впровадженням інтелектуальної інформаційно-аналітичної системи донорства крові. Це можуть бути технологічні перешкоди, фінансові витрати, адаптація до нових систем та процесів, а також потреба в навчанні персоналу та підтримці з боку відповідних організацій.

У підсумку, інтелектуальна інформаційно – аналітична система донорства крові є актуальним та перспективним рішенням для оптимізації процесу донорства, управління запасами крові та забезпечення безпеки та якості медичних послуг. Її впровадження може покращити ефективність та доступність системи донорства крові, забезпечуючи швидкий доступ до необхідної інформації, поліпшуючи координацію між різними медичними установами та забезпечуючи надійність та безпеку даних. Інформаційно – аналітична система донорства крові може сприяти збільшенню кількості активних донорів, забезпечувати постійну наявність запасів крові та зменшувати ризик виникнення кризових ситуацій, коли крові не вистачає для задоволення потреб пацієнтів.

1.2 Дослідження існуючих систем донорства крові

На сьогоднішній день в Україні існує кілька аналогів інтелектуальних інформаційно – аналітичних систем донорства крові, які спрямовані на поліпшення процесу забезпечення кров'ю та координацію донорських сеансів. Деякі з них використовуються окремими медичними установами або організаціями, а інші є національними платформами. Ось кілька прикладів аналогів інтелектуальної інформаційно – аналітичної системи донорства крові в Україні:

«Донор.ua» – це національна платформа для донорства крові в Україні, яка надає інформацію про донорські центри, акції та потреби в крові (Рис. 1.1 – Головна сторінка сайту ДонорUA). Ця інноваційна система дозволяє потенційним донорам зручно реєструватися, записуватися на прийом,

отримувати нагадування про найближчі сеанси донорства та отримувати статистичну інформацію щодо їхньої участі та внеску у донорський процес. «Донор.ua» створює зручні та ефективні механізми для взаємодії донорів із донорськими центрами, сприяючи розвитку та підтримці культури донорства в Україні.

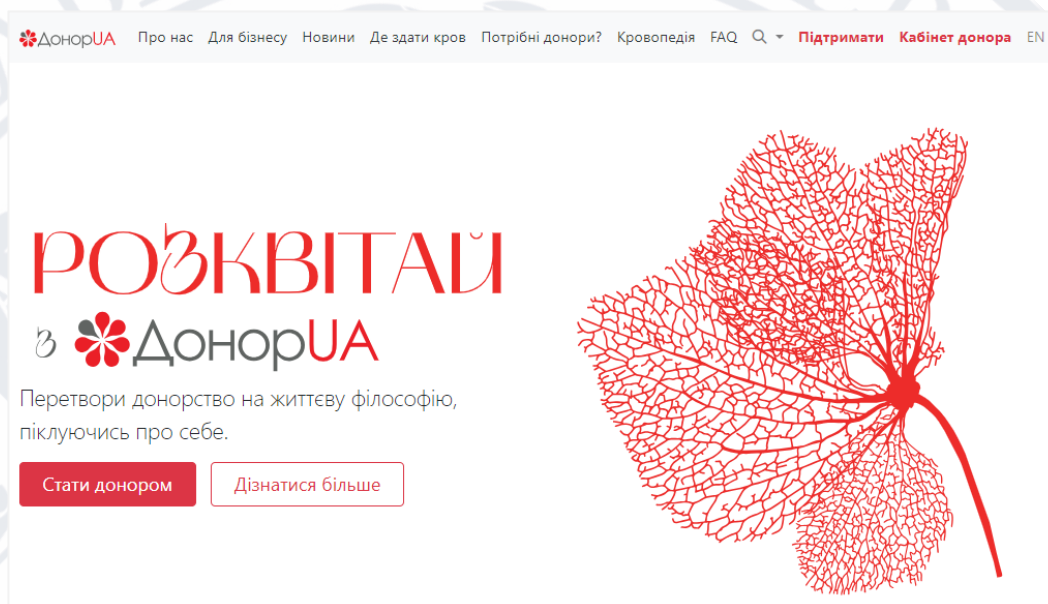


Рисунок 1.1 – Головна сторінка сайту ДонорUA

«МедДонор» – це інтегрована система донорства крові, яка використовується в деяких медичних установах України. Вона дозволяє медичному персоналу вести облік донорів, планувати донорські сеанси, здійснювати аналіз потреби в крові та відстежувати стан запасів.

«Bloodhelper» – це мобільний додаток, який допомагає залучати нових донорів та координувати донорські сеанси. Він надає користувачам зручний інтерфейс для пошуку ближчих донорських центрів, отримання інформації про вимоги до донорів та нагадування про наступний сеанс донорства.

Ці інтегровані системи донорства крові в Україні вже активно впроваджені та використовуються для поліпшення організації та координації процесу донорства крові.

Також набула чинності постанова Кабінету Міністрів України від 17 лютого 2023 року № 143, якою, серед іншого, затверджено Положення про

інформаційно-комунікаційний комплекс системи крові. Це сукупність процесів та механізмів обробки даних у сфері донорства крові та її компонентів, що функціонують за допомогою інформаційно-комунікаційних систем. Мета цього положення – забезпечити виконання функцій, пов'язаних із донорством крові та її компонентів, які визначені Законом України «Про безпеку та якість донорської крові та її компонентів крові».

Передбачено, що до інформаційно-комунікаційних систем, використовуваних в інформаційно-комунікаційному комплексі системи крові, входять система «ЄКров» та електронна система охорони здоров'я, які знаходяться на етапі проекту.

Система «ЄКров» спрямована на обробку даних та надання суб'єктам інформаційно-комунікаційного комплексу системи крові необхідної інформації відповідно до їх повноважень і функцій у галузі донорства крові та її компонентів, а також для забезпечення нормального функціонування системи крові.

Завданнями інформаційно-комунікаційного комплексу системи крові є:

- 1) забезпечення суб'єктів інформаційно-комунікаційного комплексу системи крові інформацією відповідно до їх повноважень та функцій у сфері донорства крові та компонентів крові, функціонування системи крові;
- 2) обробка даних щодо суб'єктів донорського фонду, осіб, які потребують трансфузії, та реципієнтів, а також процесів донорства та трансфузій;
- 3) інформаційне та аналітичне забезпечення процесів, таких як планування, обліку, моніторингу, контролю використання залишків крові та компонентів крові і задоволення потреб у крові та компонентах крові на загальнодержавному, регіональному та госпітальному рівні, заохочення до донорства крові та компонентів крові.

Кожна з цих систем має свої унікальні особливості та функціональність, спрямовану на підтримку медичних закладів, донорів і організацій, що займаються донорством крові. Ці системи представляють важливий крок у

напрямку автоматизації та оптимізації процесу донорства крові, що в свою чергу сприяє збереженню людських життів.

Але вони також мають свої недоліки, такі як незручність, доступність, немає можливості підлаштувати для конкретного центру або мають не потрібні функції. Наприклад, платформа «Донор.ua» містить малу або не повну базу даних донорів, що зменшує ефективність знаходження потенційних донорів, а також не має можливості реального запису до центру, а лише нагадування донору про заплановану донацію.

Отже, проаналізувавши всі аналоги, було спроектовано та розроблено зручну інтелектуальну інформаційно – аналітичну систему для донорства крові.

1.3 Постановка задачі

Створення інтелектуальної інформаційно – аналітичної системи донорства крові включає наступні аспекти:

1. Забезпечення ефективного управління донорами. Інтелектуальна інформаційно – аналітична система повинна мати здатність вести централізовану базу даних донорів, яка включає інформацію про їх медичний стан, історію донорства, контактні дані та доступність для наступних донорських сеансів. Система повинна забезпечувати автоматичне попередження донорів про можливість здати кров, нагадування про наступні сеанси донорства та розклад планування.

2. Планування та управління донорськими сеансами. Інтелектуальна інформаційно – аналітична система надає можливість планувати донорські сеанси, враховуючи наявність донорів, їх групу крові, а також потреби медичних установ та пацієнтів. Система повинна забезпечувати оптимальне розподілення донорів на сеанси та враховувати особисті переваги та обмеження донорів, такі як робочий графік та тимчасові протипоказання.

3. Прогнозування потреб у крові. Інтелектуальна інформаційно – аналітична система повинна мати можливість аналізувати дані про історію

попиту на кров. На основі цих даних система може прогнозувати майбутні потреби в крові, а також прогнозувати .

4. Сповіщення для донорів. Інтелектуальна інформаційно – аналітична система повинна надавати сповіщення донорам, про актуальні потреби у крові а також інформацію про виїзди центру донорства у міста та селища в області.

5. Забезпечення безпеки та конфіденційності. Інтелектуальна інформаційно – аналітична система повинна мати високий рівень захисту персональних даних донорів, медичної інформації та інших конфіденційних даних. Система повинна відповідати сучасним стандартам безпеки даних і мати механізми контролю доступу до інформації.

Загальна мета створення інтелектуальної інформаційно – аналітичної системи донорства крові полягає в покращенні ефективності, точності та безпеки процесу донорства крові, забезпеченні постійної наявності запасів крові та зменшенні ризиків, пов'язаних з нестачею крові. Це допоможе покращити медичну допомогу та зберегти багато життів.

Висновок до розділу 1

У висновку до огляду аналогів інтелектуальних інформаційно-аналітичних систем донорства крові в Україні можна зазначити, що на сьогоднішній день існує декілька інноваційних рішень, спрямованих на покращення організації та координації процесу донорства крові. Ці системи надають користувачам, таким як медичні установи, донори та організації, зручні та ефективні інструменти для співпраці та взаємодії.

Впровадження інтелектуальної інформаційно – аналітичної системи донорства крові допомагає вирішувати низку актуальних проблем, пов'язаних з цією сферою. Вони сприяють залученню нових донорів, підтримці існуючих донорів та координації донорських сеансів. Завдяки аналітичним функціям, ці системи надають корисну інформацію стосовно статистики, потреб у крові, стану запасів та ефективності рекламних кампаній. Крім того, вони допомагають забезпечувати безпеку та конфіденційність даних, що є важливим аспектом у такій чутливій сфері, як донорство крові.

Однак, необхідно продовжувати розвиток та вдосконалення існуючих систем. Це може включати подальшу інтеграцію з медичними інформаційними системами, розширення функціональних можливостей, покращення інтерфейсу та забезпечення широкого доступу до цих систем для медичних установ, донорів та організацій донорства крові.

Продовження інвестицій у розробку та впровадження інтелектуальних систем донорства крові в Україні може мати велике соціальне значення, зберігаючи життя та покращуючи здоров'я тих, хто потребує крові. Забезпечення ефективної, безпечної та доступної системи донорства крові є важливим кроком у розвитку медичної сфери та забезпеченні національної потреби в крові.

РОЗДІЛ 2

ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ ПРОГРАМНОГО ПРОДУКТУ

2.1 Вибір методів збору та аналізу даних

Створення інтелектуальної інформаційно – аналітичної системи для донорства крові – це важливий крок у покращенні системи збору та аналізу інформації про донорів та потреби в крові.

Нижче представлено методи збору даних, які використано в інтелектуальній інформаційно – аналітичній системі донорства крові:

1. Реєстрація донорів. Система може збирати основні персональні дані потенційних донорів, такі як ім'я, адреса, стать, вік, контактна інформація тощо. Ці дані допомагають створити базу даних донорів.

2. Медична інформація. Збір інформації про стан здоров'я донора, включаючи тимчасові протипоказання, історію донацій та інші відповідні дані, що допомагає спрогнозувати наступний візит донора та нагадати йому про це.

3. Інформація про потребу в крові. Система повинна збирати дані про потребу в крові з лікарень, клінік та інших медичних установ. Ця інформація включає групу крові, кількість необхідних одиниць, час доби та інші характеристики запиту.

4. Географічні дані. Відстеження географічного розташування донорів крові та медичних установ є важливим для ефективного розподілу крові. Географічні дані можна збирати за допомогою GPS або вводити вручну.

Вивчення інформації про донорів крові є необхідним компонентом в управлінні донорськими центрами. Цей аналіз спрямований на оптимізацію стратегій залучення та збереження донорів, а також підвищення ефективності процесів у сфері донорства. Розглянемо ключові аспекти аналізу даних донорів крові.

Аналіз даних — це процес очищення, зміни та обробки необроблених даних і вилучення актуальної інформації, яка допомагає приймати обґрунтовані

рішення. Процедура допомагає зменшити ризики, пов'язані з прийняттям рішень, надаючи корисну інформацію та статистичні дані, часто представлені у вигляді діаграм, зображень, таблиць і графіків.

Аналіз даних може мати багато аспектів та підходів, реалізовуватися за допомогою різних інструментів — в тому числі математичних, статистичних, за допомогою різноманітних способів візуалізації, але головне завдання його — стиснення інформації.

Аналіз даних дозволяє створити профілі донорів, враховуючи їхні основні характеристики, такі як вік, стать, місце проживання, група та резус-фактор крові. Це допомагає ліпше розуміти мету донорства та адаптувати стратегії привертання.

Аналіз даних дозволяє визначити групи донорів за різними критеріями, такими як регулярність донацій, тип крові, активність участі в подіях тощо. Сегментація донорів допомагає виробляти спеціалізовані стратегії для кожної групи.

Аналізуючи дані, можна виявити тенденції та патерни у донорській поведінці. Наприклад, визначити часи року, коли активність донорів може зростати або зменшуватися. Це дозволяє адаптувати маркетингові кампанії та події, розробити індивідуалізовані стратегії залучення та заохочення.

Аналіз даних також важливий для забезпечення безпеки крові та керування ризиками. Визначення можливих факторів ризику та їх моніторинг сприяє забезпеченню якості та безпеки донорської крові.

Аналіз даних донорів крові є ключовим інструментом для покращення донорських програм, забезпечення сталої популяції донорів та вирішення медичних потреб. Аналіз даних може допомогти оптимізувати розподіл крові в реальному часі, забезпечуючи її доступність для тих, хто найбільше потребує.

Для організацій, що ведуть реєстри донорів та отримувачів, важливо мати інформаційно-аналітичні системи, які допомагають автоматизувати та полегшити цей процес, забезпечуючи більшій кількості людей доступ до необхідної крові у найбільш критичний момент.

Розглянемо методи аналізу даних для інтелектуальної інформаційно-аналітичної системи донорства крові.

В інтелектуальній інформаційно-аналітичній системі донорства крові використано такі методи для аналізу даних:

1. Кластерний аналіз. Це групування донорів та медичних закладів на основі схожих характеристик. Кластерний аналіз спрямований на групування елементів в наборі даних так, щоб об'єкти всередині одного кластеру виявляли більше схожості між собою, а об'єкти, які належать до різних кластерів, були більш відмінними один від одного. Цей метод дозволяє визначити внутрішні структури та взаємозв'язки між елементами набору даних, допомагаючи виявляти приховані патерни і групи.

Основні принципи кластерного аналізу полягають в тому, що кластерний аналіз використовує метрики схожості або відмінності між об'єктами для визначення ступеня їхньої близькості. Існує кілька алгоритмів кластерного аналізу, таких як К-середніх (K-Means), ієрархічний кластерний аналіз, агломеративний метод тощо. Кожен з них має свої переваги та обмеження.

Однією з ключових задач є визначення оптимальної кількості кластерів. Це може бути визначено експертно або за допомогою певних метрик. Використання кластерного аналізу є актуальним в медицині. В медицині його можна використовувати для групування пацієнтів за спільними симптомами або результатами тестів.

Кластерний аналіз є потужним інструментом для виявлення патернів та групування даних, що допомагає в різних областях науки та бізнесу.

2. Аналіз асоціацій. Це виявлення зв'язків між різними характеристиками (наприклад, вік, група крові) та донорською активністю. Основна мета аналізу асоціацій полягає в ідентифікації часто взаємодіючих комбінацій елементів або тих, які з'являються разом в деякому контексті.

Цей аналіз може розкрити цікаві взаємозв'язки, наприклад, можливість вивчення демографічних характеристик донорів, таких як вік, стать, рівень освіти, професійна зайнятість і місце проживання. Це може допомогти зрозуміти,

які групи населення більш схильні стати донорами крові. Розуміння мотивацій донорів, а саме благодійність, свідомість, особистий досвід, та ідентифікація можливих бар'єрів, таких, як відсутність інформації, відсутність мотивації, обстеження за станом здоров'я.

Один із широковживаних методів аналізу асоціацій – це використання алгоритму Аргіогі, який ефективно визначає, як часто зустрічаються зв'язки в наборах даних. Застосування аналізу асоціацій є актуальним в медицині, де важливо виявити залежності та патерни в даних для ефективного прийняття рішень.

Моделі прогнозування. Прогнозування запитів - це процес передбачення майбутніх запитів або подій на основі аналізу історичних даних та інших доступних джерел інформації. Цей процес дозволяє приймати більш обґрунтовані рішення, готувати ресурси та вирішувати завдання заздалегідь. Аналітичні моделі можуть бути розроблені для прогнозування майбутніх запитів на кров на основі історичних даних і чинників, таких як сезонність, події чи катастрофи. Використовуючи алгоритми прогнозування, можна аналізувати історичні дані та передбачити майбутню донорську активність. Це дозволяє забезпечити належний рівень запасів крові та уникнути нестач у критичні моменти.

Для ефективного прогнозування використовуються різні методи та техніки аналізу даних, що дозволяють визначити потребу та забезпечити необхідний запас для задоволення медичних потреб. Можна застосувати алгоритми машинного навчання та статистичних методів для розробки моделей прогнозування. Моделі можуть враховувати різні фактори, такі як рекламні кампанії, свята, а також інші соціально-економічні та медичні показники.

Прогноз може бути покращений за допомогою аналізу факторів, які впливають на попит крові.

Прогнозування повинно враховувати сезонні та трендові коливання. Наприклад, взимку може збільшуватися попит через можливий зріст захворювань та аварій.

Прогнозування донацій крові є динамічним завданням, і використання різноманітних методів аналізу даних дозволяє оптимізувати управління запасами, забезпечуючи необхідну кількість крові для медичних потреб. Далі описано основні кроки та методи прогнозування запитів.

Відбувається збір даних, які стосуються запитів або подій, які плануються прогнозувати. Це можуть бути дані про користувачів, географічні дані та інше. Дані можуть містити помилки, пропуски чи аномалії, які потрібно виправити або вилучити. Визначаються фактори, які можуть впливати на запити або події, такі як сезонність, тренди, соціальні фактори, економічні показники та інші.

Далі відбувається вибір відповідної математичної моделі або алгоритму для прогнозування. Це може бути статистична модель, машинне навчання (наприклад, регресія, нейронні мережі, дерева рішень тощо) або інші методи. Дані використовуються для навчання обраної моделі. Це включає в себе підгонку параметрів моделі до фактичних даних. Перевіряється точність та ефективність моделі за допомогою тестових даних, які не використовуються під час навчання. Далі навчена модель використовується для прогнозування майбутніх запитів або подій на підставі нових вхідних даних. Порівнюються прогнози з фактичними результатами та робиться висновок якості та точності прогнозів. Важливо періодично оновлювати модель з огляду на нові дані та зміни у факторах, які впливають на запити або події.

3. Графовий аналіз: виявлення взаємодії між різними учасниками системи (лікарні, донори, лабораторії). Це процес знаходження підходящих донорів крові або органів для конкретних отримувачів на основі різних факторів і характеристик. Цей процес дуже важливий для забезпечення безпеки та ефективності трансплантації крові. Це допомагає оптимізувати ланцюг постачання та виявляти можливі точки оптимізації.

Один з найважливіших факторів відповідності донорів і отримувачів це тип крові. Донор повинен мати сумісний тип крові з отримувачем. У системі ABO і RhD існують чотири основних групи крові: A, B, AB і O, а також позитивний (+) і негативний (-) RhD-фактор. Донори і отримувачі повинні бути

вільні від антитіл та інфекційних захворювань, які можуть бути передані через трансплантацію. Також важливо враховувати розташування донора та отримувача, оскільки швидкий доступ до крові чи органів може бути життєво важливим.

Ці методи збору та аналізу даних допомагають покращити управління системою донорства крові, забезпечуючи безпеку, доступність та ефективність процесу.

2.2 Вибір інструментів для створення інтелектуальної інформаційно-аналітичної системи

В сучасному світі, де обробка та аналіз великих обсягів даних стають невід'ємною частиною успішного функціонування різноманітних організацій та підприємств, створення інформаційно-аналітичної системи стає актуальним завданням. Вибір правильних інструментів для розробки такої системи визначається рядом факторів, від технічних можливостей і потреб користувачів до вартості розробки та легкості підтримки.

У цьому розділі розглядаються ключові аспекти вибору інструментів для розробки інтелектуальної інформаційно-аналітичної системи. Починаючи з обговорення різних типів баз даних, мов програмування та фреймворків, детально розглядаються технології, які допомагають створити ефективну та потужну інформаційну систему. Поради щодо вибору оптимальних інструментів та їх інтеграції допоможуть прийняти обдумане рішення, спрямоване на досягнення поставлених цілей і вимог користувачів. Розділ також розкриває сучасні підходи до розробки та інтеграції технічного стеку.

Для оптимізації та полегшення взаємодії з донорською аудиторією було прийнято рішення створити чат-бот в Телеграмі. Чат-бот – це програма, яка автоматизує процес комунікації з користувачем через текстові або аудіо-повідомлення. Ці програми можуть функціонувати в різних месенджерах або на веб-сайтах, і їх основною метою є взаємодія з користувачем, надання інформації, виконання завдань або вирішення певних проблем.

Чат-боти можуть бути створені для різних цілей і галузей, включаючи бізнес, освіту, розваги, технічну підтримку, торгівлю тощо. Вони можуть мати різні рівні складності, від простих програм, які реагують на ключові слова або команди, до складних систем зі штучним інтелектом, які можуть розпізнавати природну мову та навіть "вивчати" на основі взаємодії з користувачами.

Розглянуто основні переваги та можливості чат-ботів. Чат-боти можуть надавати швидкі та автоматизовані відповіді на запитання користувачів. Боти можуть обробляти команди та інструкції користувачів, взаємодіючи з ними через текстові повідомлення. Деякі чат-боти можуть розпізнавати та працювати з різними мовами. Боти можуть бути інтегровані з різними сервісами чи API для виконання різних завдань, таких як робота з базою даних, відправка повідомлень тощо. Деякі чат-боти можуть використовувати технології обробки природної мови для розуміння та інтерпретації запитань користувачів. Чат-боти знаходять застосування в різних галузях, допомагаючи вдосконалити обслуговування клієнтів, автоматизувати завдання та надавати швидкий доступ до інформації.

В інтелектуальній інформаційно-аналітичній системі телеграм-бот виконуватиме роль інтерфейсу для зручного спілкування та надання необхідної інформації, а саме:

1. Реєстрація та аутентифікація. Донори можуть реєструватися та авторизуватися через телеграм-бот, забезпечуючи зручний та безпечний доступ до системи.

2. Отримання інформації. Донори можуть отримувати актуальну інформацію про потреби у крові та події або виїзди центру донорства, через бот. Це може включати новини про події, проекти або потреби в конкретних ресурсах.

3. Запис на заходи. Телеграм-бот може допомагати донорам записуватися на донацію, надаючи детальну інформацію та можливість підтвердження участі.

5. Сповіщення та нагадування. Бот може надсилати донорам сповіщення та нагадування про події, терміни, або потребу в допомозі.

Інтелектуальна інформаційно-аналітична система з вбудованим телеграм-ботом сприяє покращенню комунікації з донорською групою, роблячи взаємодію з організацією більш ефективною та зручною для обох сторін. Такий підхід може підвищити рівень участі донорів та сприяти успішній реалізації благодійних ініціатив.

Розробка чат-боту для системи донорства крові включає в себе кілька ключових етапів, спрямованих на полегшення взаємодії з донорами та покращення комунікації. Ось основні етапи розробки:

1. Аналіз потреб та визначення функцій. Проект розпочинається, проведенням детального аналізу потреб аудиторії та визначенням функцій, які буде виконувати чат-бот. Враховується можливість надання інформації про процес донорства, запис на прийом, нагадування про майбутні заходи, та інші функції, спрямовані на зручну комунікацію з донорами.

2. Вибір технічного стеку. Визначається мова програмування та фреймворки, які будуть використовуватися для розробки чат-бота. Для розробки чат-боту для донорства крові обрано такі мови програмування, як JavaScript, фреймворк ReactJS, Node.js та фреймворк Telegram Bot API та база даних MongoDB.

3. Проектування архітектури та інтерфейсу бота. Розробляється архітектура чат-боту, визначається структура та взаємодія різних компонентів, таких як обробник повідомлень, база даних, система взаємодії з користувачем тощо.

4. Підготовка майданчика на сервері. На цьому етапі готується і налаштовується серверне середовище для чат-бота, включаючи вибір платформи (хмарного сервісу або власного сервера), встановлення необхідного програмного забезпечення (веб-сервер, база даних, середовище виконання), конфігурацію безпеки, створення бази даних та інтеграцію з іншими сервісами.

5. Розробка API інтерфейсу, для інтеграції і взаємодії з ботом. На цьому етапі створюється API-інтерфейс для ефективної взаємодії та інтеграції з ботом. Це включає в себе визначення точок доступу, форматів даних, прав доступу та

інших параметрів для обміну інформацією між ботом і іншими системами чи додатками.

6. Розробка бота, призначених для користувача сценаріїв. На цьому етапі розробляються боти з визначеними сценаріями взаємодії з користувачами. Це включає в себе створення алгоритмів та логіки, яка дозволяє ботові ефективно взаємодіяти з користувачами, обробляти їх запитання, виконувати завдання та надавати необхідну інформацію.

7. Тестування функціоналу чат-бота. На цьому етапі проводиться тестування функціоналу чат-бота для виявлення та виправлення помилок, а також забезпечення оптимальної роботи. Тести включають в себе перевірку відповідей на запитання, коректність виконання завдань, взаємодію з різними типами запитань та валідацію введених даних.

8. Запуск офіційного бота в мережі. На цьому етапі відбувається запуск офіційного бота в мережі, щоб користувачі могли взаємодіяти з ним в реальному часі. Бот стає доступним для публічного використання, і організація може розпочати просування та використання бота для досягнення своїх цілей.

9. Підтримка і розвиток бота, реклама, просування. На цьому етапі бот отримує підтримку, розвивається з урахуванням потреб користувачів, та проводиться реклама для привертання нових користувачів. Здійснюється аналіз результатів та формулювання стратегій розвитку.

Веб-додаток для телеграм-бота – це інтерактивний інтерфейс, який дозволяє користувачам взаємодіяти з ботом через веб-сайт або веб-додаток, замість прямої комунікації в месенджері. Це може бути корисно для зручності користувачів та надання додаткових функцій, які не завжди доступні в телеграмі.

Користувачі можуть взаємодіяти з ботом через зручний інтерфейс веб-сайту, де вони можуть вводити дані, обирати опції, переглядати результати та інше. Веб-додаток може містити графічний інтерфейс, який поліпшує користувацький досвід, наприклад, візуальне відображення даних або використання графіків. Веб-додаток може надавати додаткові можливості, такі як перегляд історії взаємодії з ботом, налаштування параметрів, завдання або

перегляд інших статистичних даних. Важливо забезпечити безпеку та конфіденційність даних користувачів, особливо якщо через веб-додаток передається чутлива інформація. Веб-додаток повинен бути адаптований для різних пристроїв та розмірів екранів, щоб забезпечити зручність використання на різних пристроях.

Для створення веб-додатку для телеграм-бота можна використовувати різні технології, такі як веб-фреймворки, JavaScript-бібліотеки або платформи для розробки веб-додатків. Такий підхід дозволяє розширити можливості телеграм-бота і зробити його більш функціональним та доступним для широкого кола користувачів.

Тепер детальніше розглянемо вибір інструментів для розробки інтелектуальної інформаційно-аналітичної системи.

Для полегшення створення дизайну існує безліч додатків. Одним із них є Adobe Photoshop (рис. 2.1). Це програмне забезпечення, яке використовується для редагування растрових зображень, цифрового мистецтва та графічного дизайну. Він використовує багатошаровість, щоб забезпечити глибину та гнучкість процесу проектування та редагування, а також надає потужні інструменти редагування, які можна комбінувати для виконання майже будь-якої операції.

Adobe Photoshop має зрозумілий інтерфейс, багато робочих інструментів, для малювання різних форм і контурів, можливість роботи з 3D-графікою, він дуже підходить для ретуші, обробки фотографій не тільки на аматорському рівні, але і на просунутий рівень. Зображення хорошої якості (Малюнок 1). Багато дизайнерів використовують Photoshop, тому легко співпрацювати та обмінюватися файлами.

Проте навіть у цієї програми є свої недоліки, зокрема вартість преміум версії, необхідність періодичної активації ліцензії та вимога до потужного комп'ютера.



Рисунок 2.1 – Логотип програми Adobe Photoshop

Наступною програмою, яку варто розглянути, є Adobe Illustrator (рис. 2.2). Це одна з кращих програм для роботи з векторною графікою. З його допомогою можна створювати іконки, ілюстрації, банери, рекламні листівки, логотипи, макети сайтів. Він має багато функцій, таких як сенсорний тип, вільна трансформація та трансформація ляльки.

Недоліки включають різні команди для Photoshop та InDesign, часові обмеження на ліцензії та невелику криву навчання.



Рисунок 2.2 – Логотип програми Adobe Illustrator

Adobe XD представляє ідеальний інструмент у складі Adobe Creative Cloud, спеціально розроблений для веб-дизайну (рис. 2.3). Він дозволяє створювати макети для різноманітних веб-сайтів, адаптивних сторінок і веб-додатків. Основна перевага полягає у можливості створювати власні шаблони компонентів для частого використання. Крім того, проекти можна відкривати у Photoshop. Програма також надає зручну систему збереження змін і підтримує роботу з векторною графікою. Інші переваги включають імпорт проектів з інших інструментів веб-дизайну, можливість спільної роботи над проектом, створення анімацій, відео, інтерактиву та багато іншого.

Щодо цінової політики, то особисто користуватися можна безкоштовно, а для компаній є кілька варіантів покупки ліцензії.



Рисунок 2.3 – Логотип програми Adobe XD

Наступний додаток Figma (рис. 2.4). Надає онлайн-послуги розробникам і дизайнерам. Процес створення веб-сайтів стає більш доступним, завдяки можливості спільного редагування проектів дизайнерами та програмістами та миттєвому перегляду внесених змін.

Переваги програми включають багатокористувацький режим редагування, службу хмарного зберігання, створення часто використовуваних компонентів, сіток тощо, а також безкоштовний режим з обмеженою кількістю проектів.

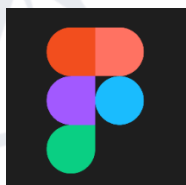


Рисунок 2.4 – Логотип програми Figma

Дослідивши та проаналізувавши всі додатки для створення дизайнів, ми зупинимося на Figma, оскільки це зручна та безкоштовна програма, яку можна використовувати як у браузері, так і встановити на локальному сервері.

Інтегроване середовище розробки (IDE) — це програмне забезпечення, створене для розробки програм, яке об'єднує різноманітні інструменти розробки в єдиному графічному інтерфейсі користувача (GUI).

Використання IDE спрощує процес створення нових програм, оскільки не вимагає ручної конфігурації та інтеграції різних утиліт під час налаштування. Розробники можуть швидко приступити до програмування, не витрачаючи годин на вивчення окремих інструментів, оскільки всі вони доступні на одному робочому просторі. Це особливо корисно для новачків у розробці, які можуть

використовувати IDE для ознайомлення зі стандартними інструментами та робочими процесами команди.

Більшість функцій IDE спрямовані на ефективне використання часу, зокрема інтелектуальне завершення коду та автоматичне генерування коду, що дозволяє уникнути введення повних послідовностей символів.

Інші широко використовувані можливості Інтегрованого середовища розробки (IDE) призначені для полегшення організації робочого процесу розробників та вирішення їхніх завдань. IDE аналізує код в режимі реального часу, визначаючи помилки, вчинені розробником, і допомагає їх виправити. Завдяки єдиному графічному інтерфейсу, розробники можуть виконувати різні дії, не перемикаючись між різними програмами. Функції підсвічування синтаксису, розповсюджені в більшості IDE, використовують візуальні підказки для відзначення граматичних конструкцій у текстовому редакторі. Деякі IDE також включають браузері класів та об'єктів, а також діаграми та ієрархії класів для конкретних мов програмування.

Існує можливість розробляти програми без використання інтегрованого середовища розробки (IDE) або, по суті, створювати власну IDE для кожного розробника, вручну інтегруючи різні утиліти з легким текстовим редактором, таким як Vim або Emacs. Для деяких розробників цей підхід привабливий через можливість ультраналаштування та контролю, які він надає. Однак в корпоративному середовищі зазвичай надається перевага економії часу, стандартизації середовища та можливостям автоматизації, які пропонують сучасні інтегровані середовища розробки.

Сучасні корпоративні команди розробників широко використовують готові інтегровані середовища розробки (IDE), які передбачено заздалегідь та найкращим чином відповідають їхнім конкретним потребам. Таким чином, основне рішення не полягає в тому, чи використовувати IDE, а в обранні оптимального варіанту IDE для конкретного випадку використання.

Існує різноманітність технічних та ділових варіантів використання Інтегрованих середовищ розробки (IDE), що також вказує на наявність на ринку

численних варіантів власних IDE з відкритим вихідним кодом. Зазвичай основними відмінними рисами між IDE є:

Кількість підтримуваних мов. Деякі IDE спеціалізуються на конкретних мовах програмування, оптимально взаємодіючи з певною парадигмою програмування.

Підтримувані операційні системи. Вибір IDE може залежати від операційної системи розробника та специфікацій цільової платформи для програми, наприклад, для Android або iOS.

Функції автоматизації. Надто різноманітні можливості IDE, включаючи рефакторинг, пошук коду та інструменти для безперервної інтеграції та розгортання, можуть бути важливими при виборі.

Вплив на продуктивність системи. Розгляд обсягу пам'яті IDE є ключовим для розробників, які планують одночасно використовувати інші пам'яті-інтенсивні програми.

Плагіни та розширення. Можливість налаштовувати робочі процеси за допомогою плагінів та розширень є додатковою перевагою деяких IDE, дозволяючи адаптувати їх до потреб розробника.

Деякі інтегровані середовища розробки (IDE) спеціалізуються на підтримці конкретних мов програмування, тоді як інші призначені для виробництва мобільних додатків, розробки веб-сайтів або орієнтовані на платформи Apple або Microsoft.

Багатомовні IDE, такі як Eclipse, Aptana, Komodo, NetBeans і Geany, розроблені для роботи з декількома мовами програмування. Наприклад, Eclipse підтримує C, C++, Perl, Python, Ruby, PHP, Java та інші. Це безкоштовне редактор із відкритим вихідним кодом, призначене для використання з різними фреймворками розробки. Хоча спочатку воно було орієнтоване на розробку на мові Java, пізніше розширило свій функціонал за допомогою плагінів. NetBeans також є багатомовним і підтримує Java, PHP, JavaScript, C, C++, Python, Ruby тощо. Це безкоштовне та відкрите середовище, а модульна структура дозволяє

додавати підтримку для інших мов програмування за допомогою встановлення додаткових модулів.

Komodo IDE є інструментом корпоративного рівня, що надає підтримку для мов програмування, таких як Perl, PHP, Python, Tcl, JavaScript, Ruby та інші. Цей продукт відзначається високою якістю, але його вартість також вища.

Artana, в свою чергу, є популярним вибором для розробників, які працюють над веб-додатками, оскільки він підтримує HTML, JavaScript, CSS, AJAX та інші мови за допомогою різноманітних плагінів.

Geany - це середовище розробки, яке підтримує C, PHP, Java, HTML, Perl, Python, Pascal та інші мови. Відзначається високим рівнем налаштування та наявністю різноманітних плагінів.

Спеціально для мобільної розробки існують IDE, такі як PhoneGap та Titanium Mobile від Appcelerator.

Багато інтегрованих середовищ розробки (IDE), зокрема ті, які підтримують декілька мов програмування, пропонують плагіни для мобільної розробки. Наприклад, Eclipse володіє такою можливістю.

Серед IDE, спрямованих на розробку HTML-додатків, особливо популярними є DreamWeaver, HomeSite і FrontPage. Ці інструменти автоматизують різноманітні завдання, пов'язані з розробкою веб-сайтів.

Хмарні IDE стають все більш важливими в розробці програмного забезпечення. Можливості таких веб-інтегрованих IDE стрімко розширюються, і через це багато великих постачальників, ймовірно, змушені внедрювати їх, щоб залишатися конкурентоспроможними на своїх ринках. Хмарні IDE надають програмістам можливість отримати доступ до свого коду з будь-якого місця.

Наприклад, Nitrous представляє собою хмарне середовище розробки, яке обслуговує Ruby, Python, Node.js і інші мови. Cloud9 IDE підтримує більше 40 мов, включаючи PHP, Ruby, Python, JavaScript (з Node.js) і Go. Heroku, у свою чергу, є хмарною платформою розробки як сервіс (PaaS), яка підтримує кілька мов програмування.

Інші IDE призначені для розробників, які працюють у середовищах Microsoft або Apple. Visual Studio, яка обслуговує VB.NET, Visual C++, C#, F#, тощо, є інтегрованою середовищем розробки (IDE) від Microsoft, призначеною для створення програм для платформи Microsoft. MonoDevelop підтримує Visual Basic, C/C++, C# та інші мови .NET.

Xcode – це інтегроване середовище розробки, яке підтримує мови програмування Swift і Objective-C, а також API Cocoa і Cocoa Touch. Використовується воно виключно для створення програм для платформ iOS і Mac. Оснащене конструктором графічного інтерфейсу та симулятором iPhone/iPad.

Espresso, натомість, спрямоване на веб-програмістів Mac і підтримує мови програмування XML, HTML, CSS, PHP і JavaScript.

Coda – інше інтегроване середовище розробки для користувачів Mac, яке підтримує PHP, CSS, HTML, JavaScript, AppleScript і Cocoa API. Ця IDE вирізняється як "розробка з одним вікном".

Є спеціальні інтегровані середовища розробки, призначені для програмістів, які працюють в певній мові програмування, такі як Jikes та Jcreator для Java, CodeLite та C-Free для C/C++, RubyMine для Ruby/Rails, і Idle для Python.

Давайте звернемо увагу на Visual Studio Code (рис. 2.5). – це редактор вихідного коду, розроблений Microsoft для операційних систем Windows, Linux і macOS. Це надзвичайно швидкий і легкий редактор, призначений для перегляду, редагування, запуску та налагодження програмного коду. В основному його використовують розробники інтерфейсів. Розглянемо деякі переваги використання Visual Studio Code.



Рисунок 2.5 – Логотип Visual Studio Code

IntelliSense для мов програмування – це загальний термін, який вказує на різні можливості редагування коду. Серед них – автозавершення коду, інформація про параметри, швидка інформація, доповнення коду, "допомога щодо вмісту" та "підказка щодо коду". У Visual Studio Code IntelliSense підтримується для мов, таких як JavaScript, TypeScript, JSON, HTML, CSS, Less і Sass. Також можна додати розширення IntelliSense для інших мов, які не входять до списку мов за замовчуванням.

Палітра команд у Visual Studio Code включає вбудований інтегрований термінал, який починається з кореня вашого відкритого проекту. Ця функція забезпечує зручність, оскільки не потрібно перемикати вікна або змінювати стан існуючого терміналу для виконання швидких команд. Використання PowerShell всередині IDE дає змогу виконувати часто використовувані завдання за допомогою ярлика. PowerShell – це оболонка з відкритим вихідним кодом і мова сценаріїв, яка базується на технології Microsoft.NET. Спрямований на допомогу технічним фахівцям, які не є розробниками програмного забезпечення, у створенні ефективних сценаріїв та інструментів для полегшення їхньої роботи. При введенні будь-якої з розпізнаваних команд, палітра команд не лише надає допомогу IntelliSense, але і автоматично запускає командний рядок для виконання завдань.

Інтегрований контроль версій у Visual Studio Code включає вбудовану інтеграцію з Git - це безкоштовна розподілена система контролю версій з відкритим вихідним кодом. Вона спрямована на роботу з проектами будь-якого розміру, що дозволяє швидко та ефективно відстежувати зміни у коді, навіть у великих проектах. Visual Studio Code працює з будь-якими локальними або віддаленими репозиторіями Git і надає візуальні символи для вирішення конфліктів перед фіксацією змін.

Налагодження є ключовою особливістю Visual Studio Code, яка пропонує вбудований налагоджувач, що сприяє прискоренню циклу редагування, компіляції та налагодження. За замовчуванням, він підтримує NodeJS і може працювати з усім, що транспілюється в JavaScript. Для інших середовищ

виконання, таких як C++ або Python, необхідно встановлювати відповідні розширення. Visual Studio Code дозволяє встановлювати точки зупинки, переглядати стек викликів та змінні під час виконання, а також призупиняти або виконувати код поетапно.

Редагування різних файлів "пліч-о-пліч" стає простішим завдяки Visual Studio Code, який дозволяє відкривати редактор "пліч-о-пліч" для одного або кількох файлів, спрощуючи взаємодію з кодом.

Інші функції керування кодом у Visual Studio Code включають мовні сервіси, такі як "Переглянути визначення", "Перейти до визначення", "Знайти всі посилання" та "Перейменувати символ". Ці функції корисні для кожного розробника і дозволяють легко навігувати та редагувати код. У Visual Studio Code також є можливість формувати код JavaScript і інших мов для поліпшення його читабельності.

2.3 Аналіз та вибір мов програмування для розробки

Розглянемо найпопулярніші мови програмування для розробки у 2023 році.

Python залишається однією з найпопулярніших мов програмування для розробки чат-ботів завдяки своїй простоті, широкій підтримці бібліотек (таких як NLTK, spaCy) і активній спільноті розробників. Python – це мова програмування, яка виокремлюється своєю простотою вивчення та використання. Її синтаксис легко читається, що робить її доступною для початківців і прискорює розробку (рис. 2.6). Зокрема, велика кількість англійських ключових слів та конструкцій роблять код на Python схожим на природну мову.



Рисунок 2.6 – Логотип мови програмування Python

Мова програмування Python відзначається широкою підтримкою та активною спільнотою розробників, що стимулює обмін досвідом, спільне вирішення завдань і постійне удосконалення проектів. Наявність різноманітних бібліотек і фреймворків, таких як Flask або Django, робить розробку чат-ботів на Python ефективною та продуктивною, сприяючи швидкому впровадженню та розвитку функціоналу.

Python відзначається підтримкою обробки природної мови (NLP). Бібліотеки, такі як Natural Language Toolkit (NLTK) та spaCy, надають потужні інструменти для аналізу та розуміння мови, сприяючи взаємодії чат-бота з користувачами.

Мова програмування Python має крос-платформений підхід, що дозволяє впроваджувати чат-боти на різних платформах та середовищах. Це надає гнучкість та масштабованість у розробці.

Python також відомий своєю підтримкою для інтеграції з іншими технологіями та сервісами, роблячи його відмінним вибором для розширення функціональності чат-бота.

Останнім часом Python відіграє ключову роль у сфері штучного інтелекту (AI) та машинного навчання (ML), надаючи розробникам інструменти, такі як TensorFlow та PyTorch, для вдосконалення роботи чат-ботів. Усе це робить Python привабливим вибором для розробки чат-ботів, особливо для тих, хто цінує простоту розробки, широкі можливості та активну спільноту.

Ще одною популярною мовою програмування є JavaScript. JavaScript, особливо в контексті використання середовища Node.js, визначається своєю універсальністю та широким застосуванням в різних областях програмування. Його можна використовувати як на стороні клієнта для розробки веб-інтерфейсів, так і на стороні сервера для побудови ефективних та масштабованих додатків.

Однією з ключових переваг є асинхронність Node.js, що дозволяє ефективно обробляти багато одночасних запитів. Це надзвичайно важливо для

чат-ботів, які повинні оперативно взаємодіяти з користувачами та обробляти повідомлення в реальному часі.

JavaScript користується великою популярністю та широким розповсюдженням, що робить його доступним для багатьох розробників. Ця багатомовність спрощує вивчення та використання мови в різних проектах (рис.2.7).



Рисунок 2.7 – Логотип мови програмування JavaScript

Багата екосистема Node.js, яка включає в себе різноманітні бібліотеки та фреймворки, забезпечує розробників інструментами для швидкої та ефективної розробки чат-ботів. Це включає в себе різні бібліотеки для роботи з популярними месенджерами та платформами, такими як Telegram чи Facebook Messenger.

JavaScript також відзначається активною спільнотою розробників, що сприяє постійному розвитку мови та підтримці для розробників чат-ботів. Загалом, JavaScript (Node.js) став популярним та потужним інструментом для створення сучасних та ефективних чат-ботів.

Наступна мова, Java, є важливою мовою програмування, також знаходить своє застосування і в розробці чат-ботів (рис. 2.8). Однією з її ключових переваг є кросплатформенність, завдяки чому програми, написані на Java, можуть запускатися на різних операційних системах без необхідності значних модифікацій. Це робить її привабливою для створення чат-ботів, які мають працювати на різних платформах та середовищах.



Рисунок 2.8 – Логотип мови програмування Java

Широкий вибір бібліотек та фреймворків для Java також грає важливу роль у її використанні для чат-ботів. Наприклад, Apache OpenNLP або DeepLearning4j дозволяють використовувати мовне програмування (NLP) та машинне навчання для поліпшення функціональності чат-ботів.

Java володіє високою масштабованістю та ефективно працює в умовах великої кількості користувачів. Це робить її ідеальним вибором для чат-ботів, які мають обслуговувати значну аудиторію.

Захист від зловживань та безпека є ще однією перевагою Java. Вона володіє вбудованою системою безпеки, що робить її популярною у галузях, де конфіденційність та захист є пріоритетом, таких як фінанси чи охорона здоров'я.

Інтегровані середовища розробки для Java, такі як Eclipse чи IntelliJ IDEA, сприяють зручному програмуванню та налагодженню. Велика та активна спільнота розробників Java забезпечує підтримку та розвиток, що є важливим аспектом у виборі мови для розробки чат-ботів.

C# (C-Sharp) — це мова програмування, розроблена Microsoft, і вона знайшла широке застосування в різних сферах, включаючи розробку чат-ботів (рис. 2.9). Однією з основних переваг використання C# є її інтеграція з платформою .NET, що відкриває доступ до багатьох бібліотек і фреймворків, таких як ASP.NET, сприяючи ефективній розробці веб-застосунків.



Рисунок 2.9 – Логотип мови програмування C#

Ця мова програмування відома своєю об'єктно-орієнтованою природою, що полегшує розробку складних систем, таких як чат-боти. Інтегровані розробницькі середовища, зокрема Visual Studio, надають зручний інтерфейс та потужні інструменти для розробки та тестування.

C# добре підтримує розробку веб-застосунків, що робить її привабливою для створення чат-ботів, які взаємодіють з веб-середовищем. Вона також сприяє побудові безпечних застосунків завдяки ініціативі Microsoft забезпечення високого рівня безпеки та захисту даних.

За допомогою .NET Core, C# став більш платформенно-незалежним, що розширює можливості використання мови для розробки чат-ботів на різних платформах. Узагальнюючи, C# є потужною мовою для розробки чат-ботів, особливо в контексті використання з .NET технологіями та інфраструктурою.

Ruby — це ще одна мова програмування, яка може бути використана для розробки чат-ботів інтерактивних систем. Привертає увагу своєю елегантністю та простотою синтаксису (рис. 2.10). Це робить її привабливою для розробників, які шукають зручний та лаконічний спосіб виразити свої ідеї. Екосистема Ruby також включає в себе різноманітні бібліотеки та фреймворки, що полегшують розробку чат-ботів.



Рисунок 2.10 – Логотип мови програмування Ruby

Принцип «Зручність перед складністю» визначає Ruby як мову, придатну для швидкої розробки та прототипування проєктів. Це особливо важливо для чат-ботів, де можливість швидко втілити та тестувати ідеї грає ключову роль.

Ruby володіє активною та сприяльною спільнотою розробників. Це означає, що розробники можуть легко знаходити підтримку, поради та готові рішення для своїх проєктів чат-ботів. Спільнота також активно вносить внески в розвиток мови та її інструментів.

Ще однією перевагою Ruby є фреймворк Ruby on Rails, який спрощує розробку веб-додатків, у тому числі тих, які взаємодіють з чат-ботами. Це робить Ruby привабливим вибором для команд, які вже працюють з Ruby on Rails та прагнуть інтегрувати чат-функціонал у свої проєкти.

PHP — це мова програмування, яка має широке використання у веб-розробці та може бути використана і для створення чат-ботів (рис. 2.11). Розглянемо деякі ключові аспекти, які роблять PHP привабливим для цієї задачі.



Рисунок 2.11 – Логотип мови програмування PHP

PHP є однією з найпоширеніших мов програмування для веб-розробки. Велика кількість веб-сайтів і систем побудована на PHP, що робить його доступним та зрозумілим для багатьох розробників. Для тих, хто вже працює з PHP, розробка чат-ботів може бути природнім продовженням їхнього досвіду.

PHP славиться своєю простотою та легкістю використання, особливо для веб-розробки. Це полегшує створення чат-ботів та їхню інтеграцію в існуючі веб-сайти та додатки. Багато розробників вважають PHP зручним інструментом для швидкої реалізації проектів.

PHP підтримує багато API та бібліотек для взаємодії з різноманітними сервісами та месенжерами, що полегшує роботу з різними платформами комунікації, такими як Facebook Messenger, Telegram та інші.

PHP має велику та активну спільноту розробників, яка надає підтримку, допомогу та регулярні оновлення. Це дозволяє розробникам швидко отримувати відповіді на свої питання та залишатися в курсі останніх тенденцій у світі PHP-розробки.

Узагальнюючи, PHP є добре відомою та широко використовуваною мовою, яка може бути успішно використана для розробки чат-ботів, особливо в контексті веб-додатків та сервісів.

Розглянувши всі вище перераховані мови, зупинимось на JavaScript.

ReactJS є відкритим фреймворком для розробки інтерфейсів користувача, який розробляється та підтримується Facebook. Цей фреймворк є одним з

найпопулярніших інструментів для створення інтерактивних веб-застосунків та веб-сайтів (рис. 2.12).

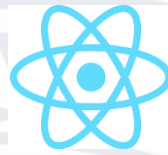


Рисунок 2.12 – Логотип Visual Studio Code

Використання віртуального DOM є ключовою оптимізаційною технікою в React. Замість традиційного прямого втручання в реальний DOM, React використовує віртуальний DOM для більш ефективного відображення змін у стані компонентів. Віртуальний DOM є легким та швидким способом реагування на зміни, спрощуючи процес маніпуляції і покращуючи продуктивність.

Щодо компонентної архітектури, React сприяє створенню додатків за допомогою компонентів, що дозволяють розбивати складні інтерфейси на компактні, автономні блоки. Кожен компонент може управляти своїм власним станом, і це полегшує розробку, роблячи код зрозумілішим та більш гнучким. Ця архітектура дозволяє ефективно розширювати та модифікувати додатки, полегшуючи їхнє управління та підтримку.

Односторінкові застосунки (SPA) легко реалізуються з використанням React, де переміщення між різними розділами відбувається без перезавантаження сторінки. Це забезпечує швидку та зручну навігацію для користувача, сприяючи високоякісному взаємодії з застосунком. Одна сторінка може динамічно змінюватися та адаптуватися до введених користувачем дій, що робить SPA ефективним та інтуїтивно зрозумілим варіантом для сучасних веб-додатків. Завдяки React, розробка SPA стає доступною і зручною, а користувачі можуть насолоджуватися плавною та безперервною навігацією між різними частинами застосунку.

React використовує JSX (JavaScript XML) для опису структури інтерфейсу користувача. JSX (JavaScript XML) - це мовний елемент, який використовується в React для опису структури інтерфейсу користувача, забезпечуючи простий і

декларативний спосіб створення компонентів. JSX введений з метою зручності розробників, надаючи їм синтаксичний цукор для більш зрозумілого та зручного взаємодії з елементами інтерфейсу.

Замість того, щоб працювати безпосередньо з методами створення DOM-елементів в JavaScript, розробники можуть використовувати JSX, щоб визначити структуру компонентів більш елегантним і легким для читання способом. Цей синтаксичний цукор полегшує процес розробки та підвищує читабельність коду, що є особливо важливим у великих проєктах.

React використовує декларативний підхід до визначення того, як повинен виглядати інтерфейс користувача при будь-якому стані. Декларативний підхід у React передбачає, що розробники описують бажаний стан інтерфейсу користувача, не задаючи конкретних кроків, як цього досягти. Замість того, щоб прописувати детальні інструкції для кожного етапу маніпуляції DOM, розробники фокусуються на описі того, що повинно статися і React бере на себе внутрішню оптимізацію та взаємодію з DOM для досягнення бажаного результату.

Цей підхід полегшує процес розробки, оскільки розробники можуть висловлювати свої наміри більш абстрактно, не занурюючись у деталі роботи з DOM. React, знаючи, що потрібно досягти, самостійно вирішує, як ефективно маніпулювати DOM для досягнення цієї мети. Це робить код більш зрозумілим, підтримуваним та менш вразливим до помилок, сприяючи кращій обслуговуваності та оптимізації в процесі взаємодії з інтерфейсом користувача.

React легко розширюється за допомогою сторонніх бібліотек та модулів. Широка екосистема React надає розробникам доступ до різноманітних інструментів та рішень, сприяючи швидкій та ефективній розробці. Завдяки цьому, розробники можуть користуватися різноманітними засобами, що полегшують їхню роботу і швидше втілювати ідеї в життя. Це відкриває можливості для створення інноваційних та високоефективних веб-додатків за допомогою розширеного функціоналу та ресурсів, доступних у розширеній екосистемі React.

В React існує активна та обширна громада розробників, яка створює доступність численних ресурсів, документації та корисних порад. Підтримка від компаній, таких як Facebook та інші, не лише гарантує актуальність фреймворка, але й забезпечує його ефективність в розвитку та вирішенні завдань. Це створює сприятливий екосистемний контекст для розробників, де вони можуть обмінюватися знаннями та знаходити відповіді на свої питання, покращуючи свої навички та роблячи процес розробки більш продуктивним.

Завдяки цим характеристикам, ReactJS продовжує залишатися ключовим інструментом у сфері розробки веб-інтерфейсів, сприяючи створенню розробниками ефективних, модульних і високопродуктивних застосунків. Цей фреймворк не лише полегшує процес розробки, але й надає потужні засоби для створення якісних та швидкодіючих веб-додатків.

Telegram Bot API представляє собою могутній інтерфейс програмування застосунків (API), який відкриває перед розробниками можливість створювати та взаємодіяти з ботами у месенджері Telegram (рис. 2.13). Цей API розширює функціональність платформи, надаючи можливість створювати різноманітні боти з різними можливостями для ефективної взаємодії з користувачами та групами. За допомогою цього інтерфейсу розробники можуть творчо реалізовувати функціональність своїх ботів та пристосовувати їх до вимог різних сценаріїв використання, щоб забезпечити зручну та інтерактивну взаємодію з користувачами Telegram.



Рисунок 2.13 – Логотип Telegram

Робота з повідомленнями є однією з ключових функцій Telegram Bot API, надаючи розробникам інструмент для взаємодії з користувачами та групами. Для оптимальної взаємодії ботів з платформою Telegram передбачено дві основні функціональності: отримання повідомлень та відправлення повідомлень.

Бот може отримувати повідомлення з різних видів чатів, включаючи приватні чати з окремими користувачами та групові чати зі спільнотами користувачів. Коли користувачі відправляють повідомлення боту, Telegram надсилає відповідний запит з інформацією про повідомлення.

Бот може взаємодіяти з користувачами, надсилаючи різні типи повідомлень. Це включає текстові повідомлення, а також мультимедійний контент, такий як фотографії, відео, аудіо та документи. Також можлива відправка різноманітних файлів, розширюючи можливості комунікації бота з користувачами.

Робота з повідомленнями надає ботам широкі можливості для взаємодії та надання користувачам багатофункціональних послуг в месенджері Telegram.

Взаємодія з користувачами є важливим аспектом в роботі ботів у Telegram Bot API. Вона включає в себе ряд функціональностей, які дозволяють ботам надавати користувачам зручні та інтерактивні можливості.

Бот може генерувати клавіатури для полегшення взаємодії з користувачами. Це може бути звичайна клавіатура або inline-клавіатура, яка виводиться прямо в повідомленні. Такий підхід робить взаємодію більш інтуїтивно зрозумілою та зручною.

Крім того, бот може відправляти inline-відповіді, які дозволяють взаємодіяти з контентом безпосередньо в повідомленнях чату. Наприклад, відправка результатів пошуку чи вибір опцій за допомогою inline-кнопок.

Взаємодія з користувачами через використання різноманітних інтерактивних елементів сприяє покращенню користувацького досвіду та робить комунікацію з ботом більш ефективною та приємною.

Робота з медіа-файлами є важливою частиною функціональності Telegram Bot API, яка дозволяє ботам ефективно обробляти та спілкуватися з різними типами мультимедійного контенту.

Бот може взаємодіяти з різними типами медіа-файлів, такими як фотографії, відео, аудіо та документи. Це дозволяє розширити можливості

обміну контентом та надає розробникам широкі можливості для створення багатофункціональних ботів.

Бот може отримувати та відправляти фотографії та відео в повідомленнях. Особливості обробки включають можливість зберігання, аналізу та подальше використання цих медіа-файлів у функціональності бота.

Telegram Bot API дозволяє ботам отримувати та відправляти аудіо-файли та різні типи документів. Це особливо корисно для створення ботів, які надають мультимедійний контент або сприяють обміну файлами.

Робота з медіа-файлами в Telegram Bot API сприяє розширенню можливостей ботів та забезпечує гнучкість у реалізації функцій, пов'язаних з обробкою та взаємодією з різними формами мультимедійного контенту.

В Telegram Bot API передбачено низку функцій, що дозволяють ботам взаємодіяти з групами та каналами, надаючи адміністративні можливості та покращуючи спілкування в спільнотах. Бот може взаємодіяти з групами та надавати адміністративні функції. Це включає в себе можливість додавати або видаляти користувачів, надавати адміністраторські права, налаштовувати параметри групи тощо.

Бот може відправляти повідомлення в групи та канали. Це дозволяє спростувати комунікацію та сповіщення великої кількості користувачів одночасно. Бот може використовувати функції модерації та фільтрації для контролю за змістом, що надсилається в групах чи каналах. Це важливо для підтримання порядку та безпеки спільноти.

Telegram Bot API дозволяє ботам отримувати статистику та аналітику щодо взаємодії з групами та каналами. Це допомагає адміністраторам зрозуміти активність спільноти та покращувати стратегію взаємодії.

Використання цих функцій дозволяє ботам ефективно адмініструвати та взаємодіяти зі спільнотами в Telegram, надаючи розширені можливості для управління та забезпечення високоякісного взаємодії з користувачами.

Telegram Bot API надає ряд функцій для відслідковування змін, що дозволяє ботам отримувати повідомлення в реальному часі та реагувати на різні події безпосередньо в чатах чи групах.

Telegram Bot API дозволяє використовувати Webhooks, що дозволяє боту отримувати повідомлення миттєво, коли вони надходять. Це реалізує механізм реального часу, спрощуючи взаємодію бота з подіями та оновленнями.

Для відслідковування змін Telegram Bot API також підтримує Long Polling. Бот може встановити з'єднання з сервером Telegram і очікувати нових повідомлень або подій. Якщо є зміни, вони відправляються боту миттєво.

Ці функції відслідковування змін дозволяють ботам ефективно реагувати на різні події, такі як нові повідомлення, приєднання чи вибуття користувачів, зміни в групах тощо. Це особливо важливо для надання користувачам актуальної та оперативної інформації.

Розглянемо деякі з найвідоміших та найефективніших бібліотек, призначених для взаємодії з Telegram Bot API. Кожна з цих бібліотек вирізняється своєю унікальною функціональністю та зручним інтерфейсом, що спрощує розробку та дозволяє розширити можливості Telegram ботів.

1. `python-telegram-bot`. Мова програмування: Python. `Python-telegram-bot` – це потужний інструмент, спеціально розроблений для роботи з Telegram Bot API на мові програмування Python. Він відзначається не лише простотою та легкістю використання, але і розгалуженим функціоналом, що надає широкі можливості для розробки різноманітних ботів. Завдяки активній спільноті розробників, ви зможете легко знаходити підтримку та зростати разом із застосуванням цієї бібліотеки.

2. `node-telegram-bot-api`. Мова програмування: Node.js (JavaScript). `Node-telegram-bot-api` – це інноваційна бібліотека, яка відкриває широкі можливості для розробників ботів в Telegram. Розроблена для використання мови програмування JavaScript і середовища виконання Node.js, ця бібліотека відрізняється не лише простим та легким інтерфейсом, але й розгалуженим функціоналом API, що дозволяє створювати ботів з різноманітним

функціоналом. Завдяки цій бібліотеці, розробники можуть зручно експериментувати та створювати інноваційні рішення для своїх Telegram ботів.

3. Aiogram. Мова програмування: Python. Aiogram представляє собою вражаючу бібліотеку для створення Telegram ботів, розроблену на мові програмування Python. Її відзначає потужний та ефективний функціонал, що забезпечує розробникам можливість використовувати асинхронний підхід для взаємодії з Telegram API. Aiogram не лише спрощує створення ботів, але й пропонує розширений набір функцій, що роблять взаємодію з Telegram динамічною та гнучкою.

Ця бібліотека розроблена з урахуванням сучасних тенденцій розробки на мові Python, що дозволяє розробникам Aiogram насолоджуватися перевагами асинхронності та багатопаровості під час створення високофункціональних Telegram ботів.

4. Telegraf. Мова програмування: JavaScript (Node.js). Telegraf є бібліотекою, яка вражає своєю легкістю та потужністю при створенні Telegram ботів на мові програмування JavaScript за допомогою Node.js. Вона виділяється своїм чистим синтаксисом та широким спектром зручних інструментів, що роблять процес розробки не тільки ефективним, а й приємним.

Telegraf відкриває перед розробниками безліч можливостей для створення високопродуктивних та функціональних Telegram ботів. Завдяки вбудованим інструментам та зручному інтерфейсу, розробники можуть швидко реалізовувати свої ідеї та додавати різноманітні функції до своїх ботів без зайвих труднощів.

Ці бібліотеки є популярними серед розробників та полегшують процес створення та управління Telegram ботами, надаючи зручний інтерфейс та високий рівень функціональності. Ми зупинимось на Telegraf.

2.4 Створення моделі інтелектуальної інформаційно-аналітичної системи

Модель - це спрощене або абстрактне представлення об'єкта або процесу в реальному світі, яке допомагає зрозуміти його функції та взаємодії. В різних галузях моделі використовуються для аналізу, прогнозування та прийняття рішень.

Математичні моделі використовують математичні формули та рівняння для опису систем та їх взаємодій, дозволяючи проводити аналіз, робити прогнози та визначати оптимальні рішення.

Комп'ютерні моделі в інформаційних технологіях використовуються для імітації реальних процесів або створення віртуальних середовищ. Це забезпечує можливість тестування та оптимізації систем без реального впровадження.

Економічні моделі в аналізі економічних явищ допомагають приймати рішення, описуючи взаємозв'язки між різними чинниками, що впливають на економічний процес.

Фізичні моделі використовуються для відтворення реальних об'єктів або систем через фізичні макети та прототипи. Це сприяє вивченню властивостей об'єктів та їх взаємодій. Загальний опис основних компонентів моделі можуть включати:

1. Вхідні дані. Цей компонент включає в себе всі дані, які подаються до системи для подальшої обробки. Вхідні дані можуть бути структурованими (наприклад, бази даних) або неструктурованими (тексти, зображення, відео, аудіо тощо).
2. Процесинг даних. Цей компонент відповідає за обробку вхідних даних, включаючи їхню очистку, перетворення та підготовку для подальшої аналітики. В цьому процесі використовуються різні алгоритми та методи обробки даних.
3. Аналітичний двигун. Ця складова включає в себе алгоритми та моделі для проведення аналізу даних.

4. Модель прийняття рішень. Цей компонент визначає, як система приймає рішення на основі результатів аналізу даних. Відповідно до поставлених завдань, система може надавати рекомендації, робити передбачення, класифікувати дані, визначати закономірності тощо.

5. Вихідні дані. Цей компонент включає в себе результати аналізу та прийняття рішень, які подаються користувачам або іншим системам. Це може бути у вигляді звітів, графіків, рекомендацій, підсумків аналізу тощо.

6. Інтерфейс користувача. Модель ПАС також повинна включати в себе інтерфейс для взаємодії з користувачами. Це може бути веб-інтерфейс, мобільний додаток або інші засоби комунікації.

7. Зберігання даних. Для збереження вхідних даних, проміжних результатів та інших інформаційних ресурсів може бути використана база даних або інші засоби зберігання.

8. Безпека та конфіденційність. Важливо враховувати аспекти безпеки та захисту конфіденційності даних у моделі ПАС. Це може включати в себе механізми автентифікації, авторизації, шифрування та інші заходи безпеки.

9. Моніторинг та підтримка. Інтелектуальна інформаційно-аналітична система повинна бути піддавана моніторингу для виявлення помилок та удосконалення її функціоналу. Також важлива підтримка для регулярного оновлення моделей та алгоритмів аналізу даних.

Інтелектуальність в інтелектуально-аналітичній системі донорства крові втілена через використання різноманітних технологій та методів для збору, аналізу та обробки інформації. Ось ці аспекти:

1. Прогностичний аналіз. Система використовує алгоритми для прогнозування попиту на кров та визначення оптимального часу для проведення кампаній з донорства.

2. Персоналізовані рекомендації. Застосування інтелектуальних алгоритмів для надання персоналізованих рекомендацій для потенційних донорів, зокрема стосовно часу та місця проведення акцій зі збору крові.

Модель інтелектуальної інформаційно-аналітичної системи (ІІАС) - це структурне та функціональне уявлення про систему, яке описує її компоненти, функції та взаємодії (рис. 2.14). Модель допомагає розуміти, як працює інтелектуальна інформаційно-аналітична система і як вона виконує завдання з обробки даних, аналітики та прийняття рішень.

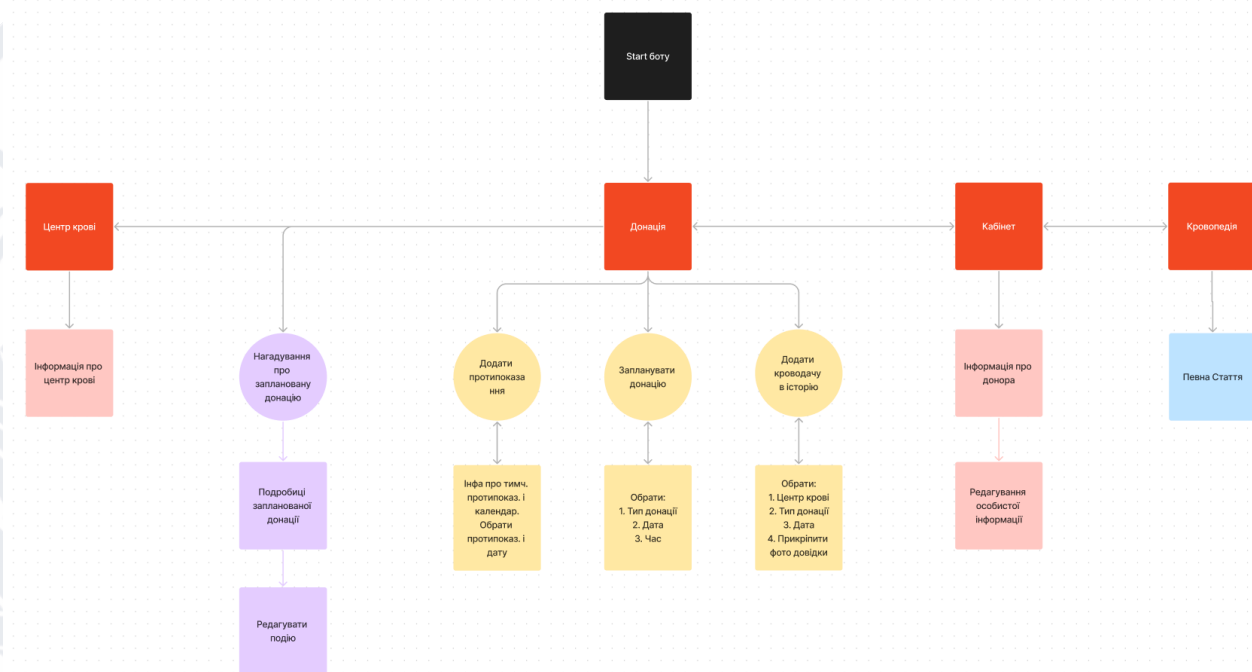


Рисунок 2.14 — Модель інтелектуальної інформаційно-аналітичної системи донорства крові

Ця інтелектуальна інформаційно-аналітична система із вбудованим телеграм-ботом має комплексну модель, яка включає в себе різні компоненти та функціональності для ефективної взаємодії з донорами. Нижче описано ключові елементи цієї системи:

1. Реєстрація та аутентифікація. Мета цього компонента полягає в забезпеченні донорам зручного та безпечного доступу до системи. У рамках цієї функціональності донори мають можливість реєструватися та авторизуватися через телеграм-бот, що забезпечує їм зручний та інтуїтивно зрозумілий інтерфейс для взаємодії з системою. Реєстрація нових донорів здійснюється безпосередньо через телеграм-бот, що спрощує та прискорює процес введення особистої інформації. Заходи щодо аутентифікації користувачів

впроваджуються для забезпечення високого рівня безпеки та конфіденційності облікових даних. Це важливий аспект, оскільки донори мають надавати особисту інформацію, і її захист є пріоритетом для системи та її користувачів.

2. Отримання інформації. Цей компонент системи спрямований на надання донорам актуальної інформації про події та потреби в крові. Однією з основних функцій є регулярне надання новин та інформації про актуальні події в сфері донорства крові. Такий підхід дозволяє донорам бути завжди в курсі останніх подій, що може викликати їхній інтерес та мотивацію для активної участі. Крім того, система передає інформацію про конкретні проекти та потреби у ресурсах, щоб донори могли бачити конкретні можливості для допомоги. Це створює прозорий механізм взаємодії між донорами та центром донорства, що сприяє ефективній координації та підвищує інформованість донорської групи.

3. Запис на донацію. Цей компонент системи спрямований на надання донорам можливості легкого та зручного запису на донаційні сесії та події, пов'язані з донорством крові. Система надає детальну інформацію про майбутні заходи, включаючи час, місце та інші важливі вказівки. Донори мають можливість не лише отримати інформацію, але і легко записатися на бажаний захід через телеграм-бот, забезпечуючи максимальну зручність та доступність процесу. Крім того, система надає можливість підтвердження участі, а також надсилає важливі вказівки та поради щодо підготовки та участі в конкретних заходах. Це сприяє ефективній організації та плануванню заходів, а також забезпечує донорам чіткі вказівки для успішної участі у донорських сесіях.

4. Сповіщення та нагадування. Цей компонент системи відіграє важливу роль у забезпеченні ефективної комунікації з донорами, щоб повідомляти їх про ключові події, терміни та можливості для допомоги. Система взаємодіє з донорами, висилаючи сповіщення та нагадування про надходження подій, таких як донорські сесії, благодійні заходи чи інші активності, пов'язані з донорством крові. Крім того, вона надає індивідуальні сповіщення, зокрема після реєстрації чи запису на конкретні заходи. Це дозволяє створити персоналізований підхід до кожного донора, надаючи йому актуальну та цікаву

інформацію, а також нагадуючи про важливі події, які можуть зацікавити та мотивувати до участі.

В цілому, цей компонент сприяє підтримці та утриманню інтересу донорів, роблячи їхню участь більш свідомою та взаємодією з системою більш динамічною. Ця модель дозволяє створити зручний та ефективний інструмент для взаємодії з донорською групою, сприяючи підвищенню рівня участі та покращенню комунікації між донорами та організацією.

Ця модель надає загальне уявлення про структуру і функції інтелектуальної інформаційно-аналітичної системи. Варіації можуть існувати в залежності від конкретного завдання та ресурсів, які доступні для розробки системи.

Висновок до розділу 2

Цей розділ присвячено вивченню різних інструментів і мов програмування, які можна використовувати для розробки веб-додатків для інтелектуальних систем аналізу та інформації про донорство крові.

Під час аналізу було розглянуто кілька популярних інструментів, таких як фреймворки для веб-розробки, бази даних і системи управління базами даних, інструменти для фронтенд-розробки та інші. Були проаналізовані їх переваги, недоліки, функціональні можливості та придатність для реалізації поставлених завдань.

Під час аналізу було встановлено, що реалізація інтелектуальної інформаційно-аналітичної системи донорства крові можна створити за допомогою сучасних технологій і інструментів. Технології, такі як Node.js, мають потужність та швидкодію для серверної частини додатку.

У виборі мови програмування JavaScript виявилася переважаючою серед розробників через свою простоту, зручність читання коду та наявність багатофункціональних бібліотек. Аналіз інструментів та мов програмування для створення веб-додатку свідчить про наявність різноманітних варіантів для реалізації інтелектуальної інформаційно-аналітичної системи донорства крові. Вибір конкретних інструментів і мов програмування залежить від вимог проєкту, його складності, ефективності та масштабованості системи.

Необхідно врахувати такі фактори, як наявні ресурси, інтеграцію з іншими системами, підтримку спільноти розробників та зручність використання. Важливо забезпечити високу продуктивність системи, надійність, безпеку та зручний інтерфейс для користувачів.

У підсумку, було зроблено обґрунтований вибір, зважаючи на потреби проєкту, технічні можливості та функціональні вимоги.

РОЗДІЛ 3

РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ДОНОРСТВА КРОВІ

3.1 Аналіз вимог до системи та її функціональні можливості

Структура роботи інтелектуальної інформаційно-аналітичної системи донорства крові включає наступні компоненти (рис. 3.1):

1. Система отримує вхідні дані, які включають інформацію про донорів, медичні записи, запаси крові, розклади та результати аналізів.
2. Система виконує обробку та аналіз вхідних даних, використовуючи різні алгоритми та методи. Це може включати класифікацію донорів за певними характеристиками, прогнозування попиту на кров, виявлення потенційних донорів тощо.
3. Система зберігає дані у базі даних, що дозволяє ефективно управляти та доступ до інформації про донорів, медичні записи, розклади збору крові та інші важливі дані.
4. Система має веб-інтерфейс, який дозволяє користувачам взаємодіяти з системою. Це може включати реєстрацію нових донорів, перегляд і оновлення особистої інформації, перегляд розкладу збору крові, запитання та відповіді з медичним персоналом тощо.
5. Система може надавати рекомендації користувачам щодо можливостей для здійснення донорства крові на основі їхніх характеристик та попереднього історичного аналізу. Також, система може надсилати оповіщення користувачам про нагадування щодо планового збору крові або випадкових потребах в крові.
6. Система надає можливість проводити аналітику даних. Це допомагає в оцінці ефективності системи донорства крові, відстеженні тенденцій та прийнятті обґрунтованих рішень для поліпшення процесу.

Коли користувач запускає телеграм-бот, він вітається та надає можливість зареєструватися або авторизуватися для отримання індивідуалізованої інформації. Під час реєстрації користувач вводить свої дані аутентифікації, такі як номер телефону та пароль. Бот виводить основні особисті дані користувача, такі як ім'я, прізвище та група крові, та просить підтвердити або виправити їх за потреби. Одночасно бот проводить перевірку, чи містить база даних лікарні медичну інформацію користувача.

Цей етап є важливим для забезпечення точності та актуальності медичних даних користувача в системі, а також для забезпечення безпеки та конфіденційності цієї інформації.

Система регулярно надсилає користувачам повідомлення щодо актуальних потреб у крові, включаючи інформацію про виїзд донорського центру в їхню локацію або поблизу неї. Крім того, користувачі отримують повідомлення про майбутні заходи донорства, включаючи виїзди донорських центрів, що дозволяє їм вчасно підготуватися до участі.

Бот систематично повідомляє користувачів про основні правила та рекомендації для безпечного проведення донорських сесій. Крім того, користувачі мають можливість обирати конкретний захід для донорства крові та записуватися на нього, отримуючи підтвердження успішного запису. Перед самою подією бот автоматично надсилає нагадування за день чи два, щоб донор міг ефективно планувати свій час та підготуватися до донорської сесії.

Також система надсилає повідомлення про розклад роботи донорського центру, щоб користувачі були завжди інформовані про доступні часи для донорства. Нагадування про правила та рекомендації для безпечного донорства регулярно надходять користувачам, щоб підтримувати свідомість та дотримання необхідних вимог для їхнього власного здоров'я та безпеки.

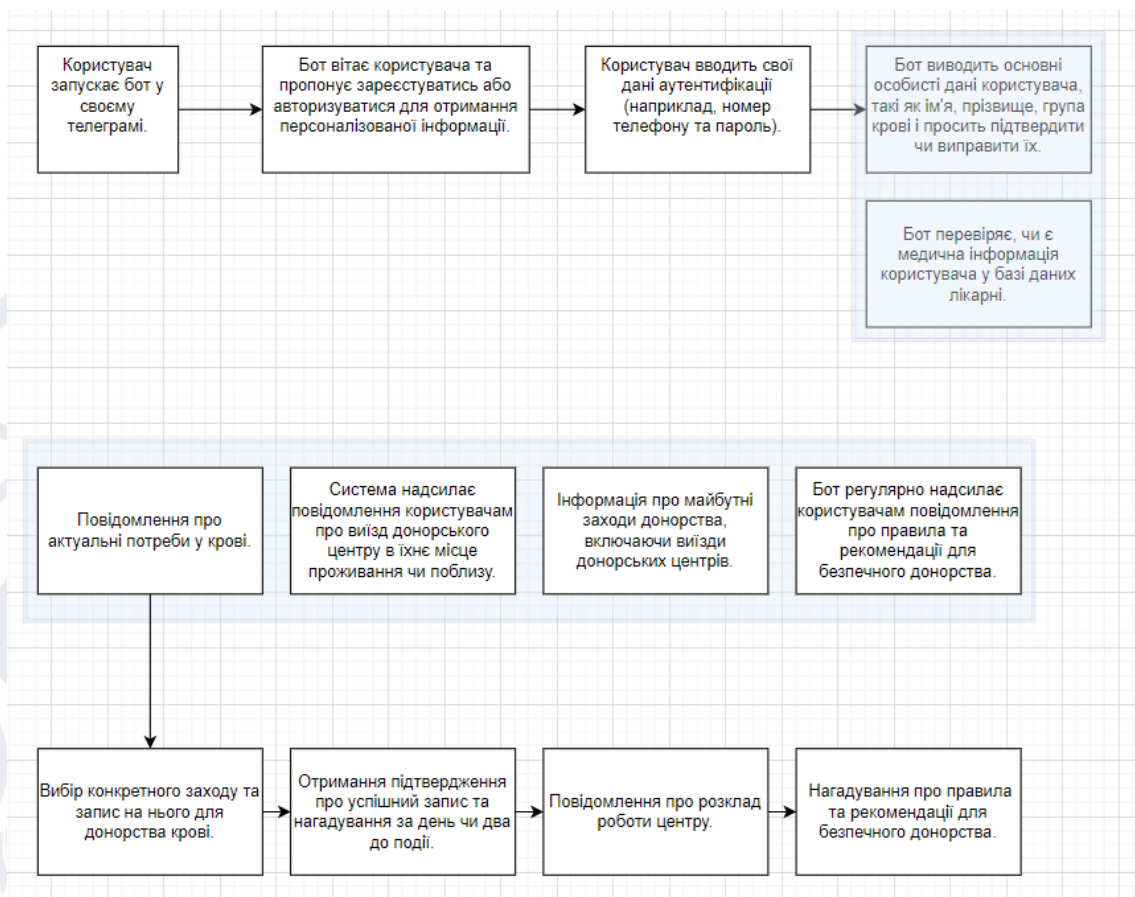


Рисунок 3.1 — Структура роботи телеграм-боту для донорства крові

Загалом, інтелектуальна інформаційно-аналітична система донорства крові має структуру, що об'єднує вхідні дані, обробку та аналіз, базу даних, веб-інтерфейс, рекомендації та оповіщення, а також звіти та аналітику. Ця структура дозволяє ефективно управляти процесом донорства крові, сприяє збору та аналізу необхідної інформації, а також забезпечує зручну взаємодію між користувачами та системою.

Інтелектуальна інформаційно-аналітична система донорства крові має великий потенціал у поліпшенні ефективності та якості донорських програм, сприяючи збереженню життя та покращенню здоров'я людей.

3.2 Створення бази даних та її структура

Бази даних стали невід'ємною частиною інфраструктури для забезпечення функціональності безлічі програм і систем у різних сферах. Їх широке

застосування охоплює бізнес, освіту, науку, медицину та інші галузі, визначаючи ключовий фундамент для розвитку і ефективності в цифровому суспільстві.

В сучасному інформаційному ландшафті, бази даних використовуються для зберігання та обробки великих обсягів різноманітних даних. Вони дозволяють організаціям здійснювати зручний та швидкий доступ до інформації, а також використовувати аналітичні інструменти для отримання цінних інсайтів.

Без ефективного управління даними навіть найперевіреніші технології та програми можуть стикатися з труднощами в їх функціонуванні. Таким чином, бази даних відіграють ключову роль у розвитку та стабільності сучасного інформаційного суспільства.

База даних є фундаментальним елементом сучасних інформаційних технологій, який використовується для ефективного зберігання та організації великого обсягу даних. Вона виступає як центральне сховище, де можна зберігати, оновлювати, видаляти та витягувати інформацію, спрощуючи роботу з даними для користувачів та розробників.

Таблиці виступають у ролі основних структурних компонентів в базах даних. Кожна окрема таблиця визначає конкретний вид даних та включає стовпці, які відображають конкретні атрибути чи характеристики. Ця структура визначає організацію інформації та надає основну схему для зберігання та організації даних в базі даних.

У кожній таблиці кожен рядок представляє собою унікальний запис, який може містити різні типи інформації. Кожен стовець визначає конкретну властивість даних, надаючи системі структурованість та логічність в організації великої кількості інформації в базі даних.

Таким чином, таблиці є ключовим елементом, що визначає схему та структуру бази даних, забезпечуючи ефективне управління та доступ до збережених даних.

Запити представляють собою ключовий механізм для взаємодії з базою даних, надаючи здатність отримувати та змінювати інформацію за визначеними

умовами. Використання запитів в базі даних дозволяє розробникам та користувачам отримувати точно ті дані, які їм необхідні для конкретних завдань.

Запити можуть виконувати різні функції, такі як витягування даних із певної таблиці чи групи таблиць, фільтрація результатів за певними умовами, сортування даних за певними критеріями, а також зміна та оновлення інформації в базі.

Цей механізм стає ключовим для того, щоб зробити доступ до даних ефективним та оптимізованим. Запити в базі даних надають інструмент для точного взаємодії з інформацією, забезпечуючи ефективний та контрольований доступ до великих обсягів даних.

Рядки (записи) або кортежі є елементами даних у таблицях бази даних. Кожен рядок відображає конкретний комплект інформації, пов'язаний із певним об'єктом чи сутністю.

Ці записи містять набір атрибутів або полів, які описують різні характеристики об'єкта чи сутності. Іншими словами, рядки в таблиці бази даних представляють конкретні екземпляри даних, а кожен рядок має свою унікальну ідентифікацію та набір значень, які визначають його стан.

Ключові поля визначають унікальність записів у таблицях бази даних. Ці поля можуть виступати у ролі первинних ключів, що ідентифікують унікальний рядок в межах однієї таблиці, або зовнішніх ключів, що встановлюють зв'язки між різними таблицями.

Первинні ключі служать для унікальної ідентифікації кожного запису в межах таблиці, забезпечуючи унікальність та структурованість бази даних. Зовнішні ключі визначають зв'язки між різними таблицями, де значення одного ключового поля утримує посилання на конкретний запис в іншій таблиці, покращуючи організацію та зв'язок між даними.

Типи баз даних визначаються згідно з їхнім логічним та фізичним організаційним принципом.

Реляційні бази даних є важливою складовою світу управління даними. Вони використовують концепції реляційної моделі баз даних, що сприяє їхній

широкій популярності та використанню. У цих базах даних інформація організована у вигляді таблиць, кожна з яких відображає певний тип даних чи об'єкт. Ключовим елементом реляційних баз є зв'язки між таблицями, що реалізуються за допомогою ключів, які визначають стосунки та спільні атрибути між різними наборами даних.

Однією з основних особливостей є жорстка структура даних. Реляційні бази даних вимагають попередньо визначених схем та типів даних для кожного стовпця, що дозволяє системі ефективно керувати інформацією та забезпечувати її консистентність. Найпоширенішими представниками реляційних баз даних є системи, такі як MySQL, PostgreSQL та Microsoft SQL Server, які використовуються в різних галузях для забезпечення надійного та структурованого управління даними.

Інтернет-орієнтовані бази даних відкривають нові можливості для зберігання та обміну даними, активно використовуючи потужності мережі Інтернет. Вони вирізняються можливістю взаємодії та обміну інформацією через веб-служби та API, що робить їх ідеальними для застосувань, де важлива віддалена робота з даними.

Однією з ключових переваг є легкий доступ до даних з будь-якого місця через Інтернет. Це сприяє зручності та швидкості обміну інформацією між різними користувачами та системами. Зокрема, веб-служби дозволяють взаємодіяти з базою даних безпосередньо через HTTP-протокол, що розширює можливості інтеграції та розробки додатків.

Деякі популярні представники цього типу баз даних включають Amazon DynamoDB, Firebase та Google Cloud Firestore, які демонструють велику ефективність в управлінні даними в середовищі Інтернету.

Також існують, NoSQL бази даних, які представляють сучасний підхід до управління даними, використовуючи альтернативні моделі даних, такі як документи, граfi чи ключ-значення. Ці бази даних здебільшого використовуються для зберігання та обробки неструктурованої інформації, що

робить їх ідеальними для сучасних завдань, де дані можуть мати різну структуру та формат.

Однією з ключових особливостей NoSQL баз є їхня гнучкість та здатність пристосовуватися до змін в обсягах та типах даних. Вони дозволяють зберігати та обробляти великі обсяги даних ефективно, що робить їх популярними для використання в розподілених системах та великих проєктах з обробкою даних в реальному часі.

Серед популярних NoSQL систем можна виділити MongoDB, Cassandra, та Redis, які забезпечують розробникам широкі можливості для зберігання та оптимізації роботи з різними видами даних.

Переходимо до розгляду конкретної бази даних – MongoDB. Ця база даних відзначається своєю гнучкістю та можливістю роботи з неструктурованими даними. Давайте поглибимось у її особливості та переваги для розуміння, як вона доповнює роль баз даних у сучасному світі.

MongoDB — це визнана та широко використовувана система управління базами даних, яка виділяється своєю гнучкістю та здатністю ефективно опрацьовувати неструктуровані дані.

На відміну від звичайних реляційних баз даних, MongoDB використовує модель документів, де дані зберігаються у вигляді документів у форматі BSON (Binary JSON). Ця архітектура надає розробникам велику свободу в обробці та організації різноманітних типів інформації. Розглянемо основні особливості MongoDB.

Однією зі значущих переваг є гнучкість схеми, оскільки MongoDB не вимагає строгої схеми для даних. Це означає, що можна легко працювати з різноманітними та змінюючимися даними без необхідності визначення жорсткої структури.

MongoDB також славиться своєю здатністю до горизонтального масштабування, що робить можливим розподіл даних на різні сервери для ефективного використання ресурсів, особливо в проєктах з великим обсягом даних.

Агрегаційні фрейми, вбудовані в MongoDB, надають можливість виконувати потужні операції обробки та аналізу даних безпосередньо в базі даних, що полегшує розробку та оптимізацію даних.

Окремо варто відзначити підтримку геопросторових запитань, яка вбудована в MongoDB. Ця функціональність стає особливо важливою для додатків, що взаємодіють з геолокаційною інформацією, дозволяючи ефективно опрацьовувати географічно розподілені дані.

MongoDB відкриває безліч можливостей для розробників завдяки своїй гнучкості та здатності легко масштабуватися для обробки зростаючого обсягу даних. Система надає розробникам велику свободу у визначенні та організації структури документів, що робить її ідеальним вибором для проектів з різноманітними вимогами до даних.

Завдяки своїй структурі документів та оптимізованим запитам, MongoDB гарантує не лише швидкий доступ до даних, але й високий рівень ефективності. Індексація та керування пам'яттю в базі даних сприяють оптимізації продуктивності для високонавантажених додатків.

MongoDB може розраховувати на активну та велику спільноту розробників, що забезпечує доступ до різноманітних ресурсів, документації та допомоги. Це робить використання та вдосконалення бази даних більш приємним та ефективним завдяки обміну досвідом та рішеннями спільноти.

MongoDB постійно впроваджує новаторські рішення та технології, що дозволяє розробникам використовувати передові можливості в сфері обробки даних. Застосування сучасних інновацій сприяє підтримці та просуванню проектів в різних галузях розробки програмного забезпечення.

Здатність працювати з різноманітними типами даних, горизонтальне масштабування та підтримка геопросторових запитань роблять MongoDB перспективним вибором для проектів, які вимагають високої продуктивності та швидкодії.

Проектування бази даних є важливим етапом у розробці інформаційних систем, що дозволяє створити ефективну та надійну структуру для зберігання та управління даними.

Проектування бази даних - це процес створення структури та схеми для зберігання та управління даними в інформаційній системі. Правильне проектування бази даних допомагає забезпечити ефективне зберігання, організацію та доступ до даних.

Цей процес складається з кількох етапів, кожен з яких визначає концепції та вимоги до бази даних.

Перший етап передбачає вивчення вимог користувачів та бізнес-потреб для розробки бази даних. Аналізуються типи даних, їх обсяг, взаємозв'язки, інтеграція та інші фактори.

На другому етапі розробляється концептуальна модель, яка відображає загальні концепції та взаємозв'язки між різними сутностями. Використовуються інструменти, такі як ER-діаграми.

Логічна модель бази даних визначає структуру даних на рівні таблиць, відносин та атрибутів. Зазвичай використовуються схеми даних, які визначають об'єкти бази даних та їхні взаємозв'язки.

Далі визначається, як дані будуть зберігатися фізично на носіях. Розробляються таблиці, індекси, ключі, вибір системи керування базами даних (СКБД).

Процес нормалізації допомагає уникнути аномалій та підтримує цілісність даних. Таблиці розбиваються на менші, що допомагає уникнути дублювання інформації.

Створення запитів, які дозволяють отримувати, оновлювати, видаляти та вставляти дані. Розробка зберігальних процедур для оптимізації взаємодії з базою даних.

Виконання оптимізації запитів, індексації та інших методів для підвищення продуктивності. Проведення тестування для перевірки правильності та ефективності роботи бази даних.

Таблиця «Донори» в базі даних зберігає повний набір інформації про кожного донора, що є важливою складовою для впорядкування та ефективного управління процесами донорства крові. Кожен донор має призначений унікальний ідентифікатор, який відокремлює його від інших учасників системи.

Особисті дані донора включають його ім'я та прізвище, групу крові та резус-фактор – важливі параметри, які визначаються для забезпечення безпеки та сумісності при процедурі донорства. Крім того, таблиця зберігає контактні дані донора, такі як номер телефону та місто проживання, що дозволяє ефективно організувати та спрямовувати інформацію до донорів.

Особисті характеристики, такі як дата народження, стать, зріст та вага, важливі для визначення придатності донора та створення індивідуального підходу до кожного учасника процесу. Ця таблиця є фундаментальним ресурсом для ведення обліку та аналізу інформації про донорів, забезпечуючи необхідну базу для організації та управління донорською діяльністю.

Таблиця «Збір крові» в базі даних відображає повний контекст інформації, пов'язаної з процедурою збору крові від кожного донора. Кожен запис у цій таблиці має свій унікальний ідентифікатор, що дозволяє однозначно ідентифікувати кожну процедуру збору крові.

Ключовим атрибутом цієї таблиці є зв'язок із конкретним донором, що визначає, від якого саме донора була отримана кров. Цей аспект важливий для створення повної історії донорства та забезпечення точності інформації.

Таблиця також фіксує дату здачі крові, що є ключовою інформацією для ведення календаря та планування подальших процедур. Крім того, фіксується місто, де був проведений збір крові, і тип здачі крові, що може включати різні види процедур чи специфікації збору.

Ця таблиця є ключовим ресурсом для ведення обліку та координації процесу збору крові, забезпечуючи цілісність та структурованість даних в рамках системи управління донорською діяльністю.

Таблиця «Центр крові» містить компренсивну інформацію про кожен центр збору крові. Кожен запис у цій таблиці має унікальний ідентифікатор, що

дозволяє однозначно ідентифікувати кожен центр крові та забезпечує зручний доступ до його інформації.

Основні атрибути таблиці включають назву центру крові, контактну інформацію, яка включає номер телефону та електронну пошту, місце розташування центру, а також графік його роботи. Ця інформація є ключовою для ефективної координації та співпраці між центрами крові та їхніми потенційними донорами. Графік роботи центру крові визначає часи, протягом яких можна здати кров, що дозволяє донорам вчасно планувати та здійснювати свої візити.

Таблиця «Банк крові» виконує важливу роль у системі управління донорством крові, зберігаючи облік інвентарю доступної крові для подальшого використання. Кожен запис у цій таблиці має унікальний ідентифікатор та пов'язаний з конкретною партією або зібраною порцією крові.

Основні атрибути таблиці включають групу крові та резус-фактор, що визначають тип крові та його сумісність, а також кількість наявної крові в мілілітрах. Ця інформація є важливою для ефективного управління запасами крові, дозволяючи точно визначати наявні резерви та вчасно реагувати на потреби пацієнтів.

Таблиця «Банк крові» допомагає підтримувати збалансований та досконалий запас крові, сприяючи надійному функціонуванню системи збору та використання крові в медичних установах та центрах крові.

На рисунку 3.2 наведена структура бази даних для інтелектуальної інформаційно-аналітичної системи донорства крові.

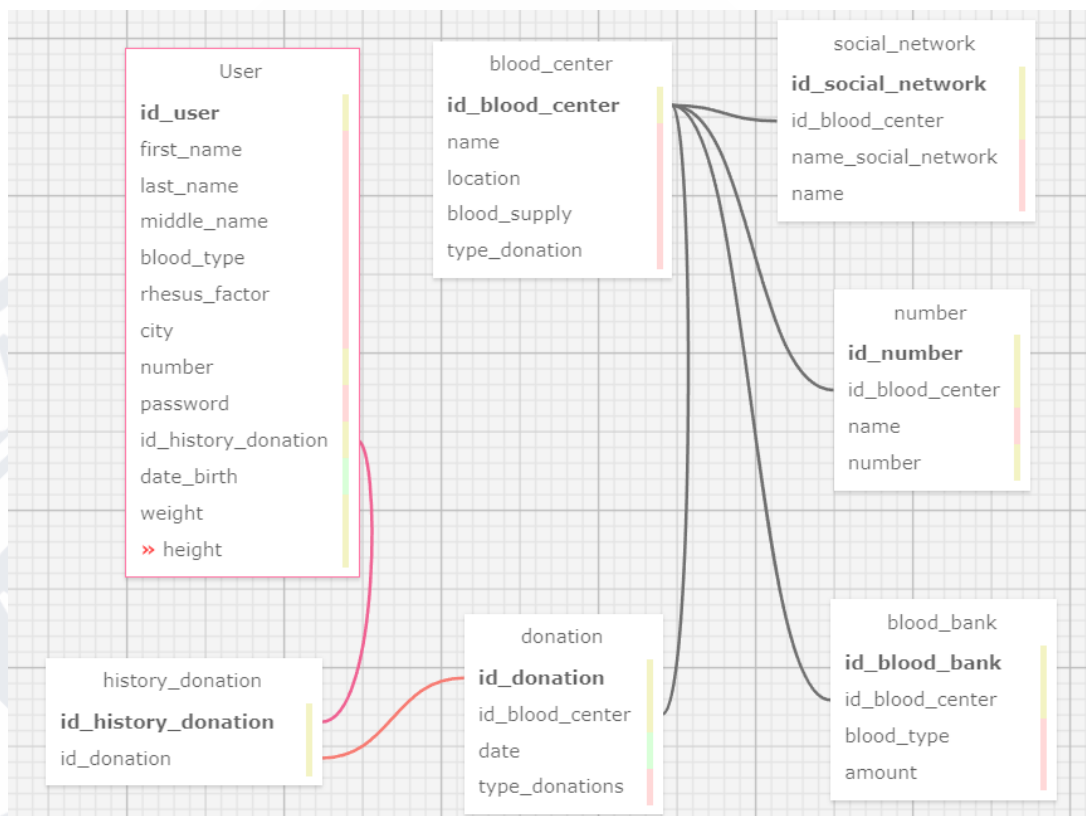


Рисунок 3.2 – Структура бази даних для інтелектуальної інформаційно-аналітичної системи донорства крові

Ця база даних дозволяє відстежувати донорів, їхні здачі крові та наявний інвентар крові. За допомогою цієї системи можна вивчати інформацію про донорів, контролювати процеси здачі крові та відслідковувати наявний запас крові для подальшого використання у медичних потребах.

3.3 Програмна реалізація

Розглянемо телеграм бот із вбудованим додатком для донорства крові, можна визначити, що він відзначається високою функціональністю та інноваційним підходом до організації донорських процесів в місті Вінниця. Створений з метою полегшення та покращення доступності донорства, бот виступає важливим інструментом для активізації цього благородного процесу. Бот забезпечує зручний інтерфейс та інформаційні ресурси для донорів, координацію донорських центрів та відстеження результатів донорства.

На рисунку 3.3 зображено логотип та загальна інформація про бот.

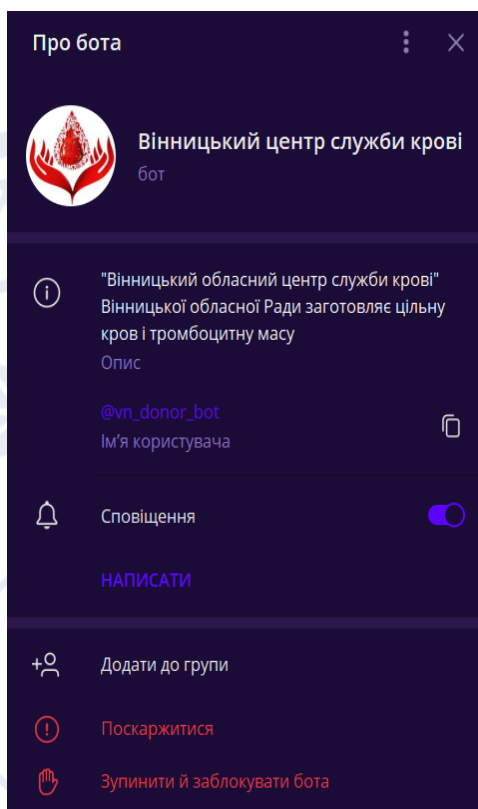


Рисунок 3.3 – Логотип та загальна інформація про бот

Розглянемо основні функціональні можливості телеграм-бота.

1. Реєстрація або авторизація користувача:

- Користувачі можуть легко зареєструватися або виконати вхід для доступу до особистого облікового запису.
- Профіль користувача стає зручним інструментом для зберігання та керування основною інформацією, такою як група крові, контактні дані, дата останньої донорської крові, а також інша суттєва інформація.

2. Контакти та загальна інформація про центри крові (рис. 3.4):

- Користувачі можуть легко отримати необхідну інформацію про центр крові, таку як їхні місцезнаходження, години роботи та контактні дані.
- Бот надає зручний доступ до загальної інформації про послуги, які надаються в центрі, такі як види донорських сеансів, процедури та інші корисні відомості.

- Користувачі можуть здійснювати зворотній зв'язок або задавати питання, використовуючи контактні дані центру крові, що дозволяє їм отримати детальну консультацію або уточнення стосовно донорства крові.
- Всі необхідні контакти та інша інформація доступні в одному місці, сприяючи легкості взаємодії та взаєморозуміння між користувачем та центром крові.

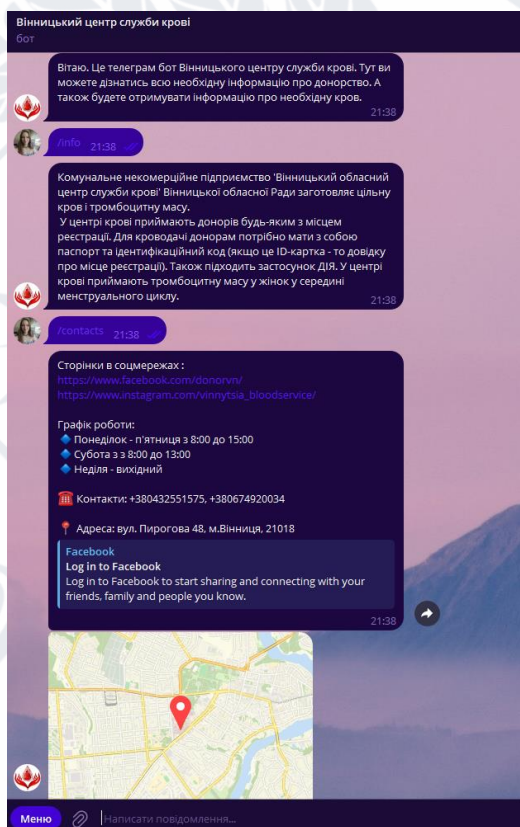


Рисунок 3.4 – Контакти та інформація про центр крові

3. Запис на донорські сеанси (рис. 3.5):

- Користувачі мають можливість вибрати оптимальний час для участі в сесії донорства, використовуючи доступний розклад, та легко здійснити реєстрацію на обраний сеанс.
- Бот автоматично висилає нагадування, щоб забезпечити своєчасне сповіщення про наближення запланованої донорської сесії.
- Підтримка користувачів у виборі ідеального часу для донорства робить процес участі максимально зручним та ефективним.

- Зареєстровані учасники отримують повідомлення, яке дозволяє їм приготуватися до сесії та надає всю необхідну інформацію щодо місця та часу донорського сеансу.

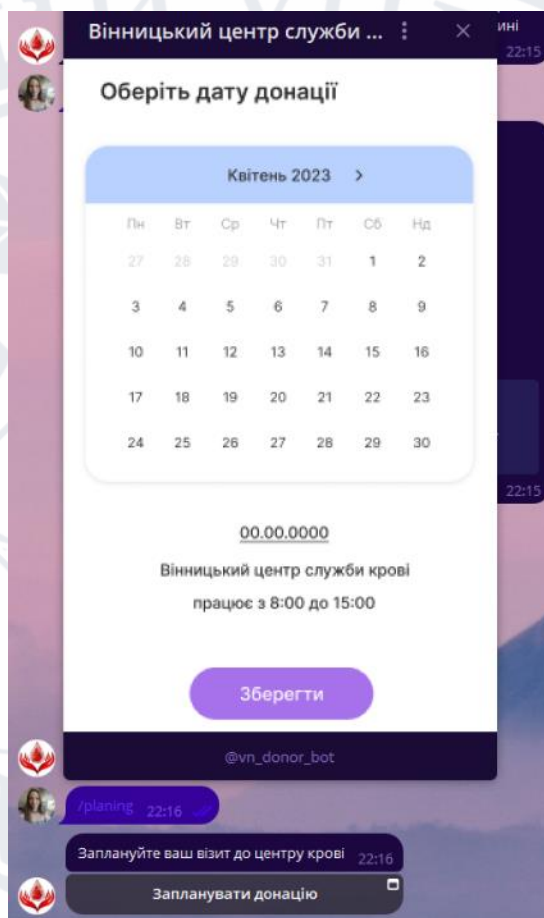


Рисунок 3.5 – Екран з плануванням донації

4. Інформаційний розділ (рис. 3.6):

- Надання ключової інформації щодо процесу донорства крові, критеріїв допуску для донорів, а також визначення переваг та можливих ризиків цього важливого процесу.
- Детальні інструкції щодо підготовки до донорського сеансу, а також розповсюдження популярних запитань і відповідей, що сприяє належному розумінню та забезпечує комфорт користувачів.
- Цей розділ дозволяє користувачам глибше ознайомитися з усіма аспектами донорства крові, роблячи акцент на прозорості та повному розумінні учасниками усіх аспектів процедури.

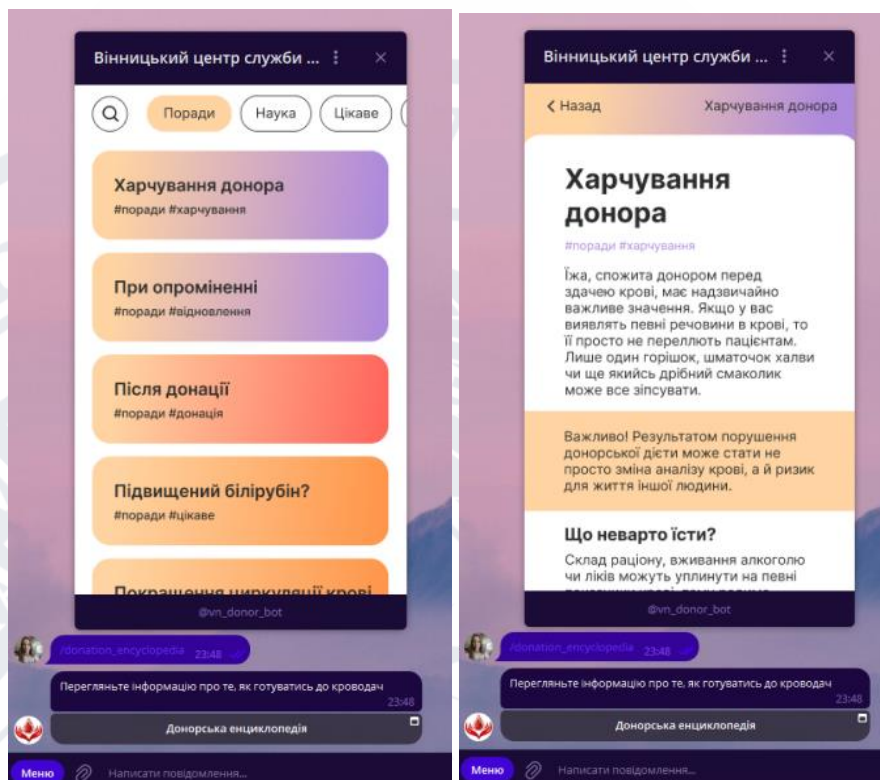


Рисунок 3.6 – Екран з статтями

5. Збереження останньої дати донації:

- Система нагадувань та сповіщень про наближення очікуваних донорських сеансів, що дозволяє користувачам завчасно планувати та приготуватися до участі.

Розроблений телеграм-бот має величезний потенціал для подальшого вдосконалення та розвитку. У межах даної роботи був реалізований продукт з базовим набором функцій, спрямованих на забезпечення стабільної роботи та демонстрації результатів магістерського дослідження. Щоб забезпечити його доступність в мережі Інтернет, необхідно лише розмістити веб-додаток на сервері та налаштувати доменне ім'я.

Процес покращення та розширення функціональності телеграм-боту може включати наступні етапи:

1. Можливість перегляду особистої історії донорських сеансів, включаючи дати, типи, що були надані під час кожної донації.

2. Інтеграція зі штучним інтелектом (AI) і машинним навчанням (ML), що може покращити прогнозування та автоматизовано формувати індивідуалізований графік майбутніх сеансів донорства, яке враховує частоту та різноманіття видів донорської участі. Крім того, система автоматично підлаштовує графік відповідно до історії донорської активності, забезпечуючи максимальну зручність та участь донора.

3. Можливість зберігати документи з аналізами крові і результати донорств, а також надавати користувачам інформацію про те скільки життів він врятував, аби мотивувати на наступні кровоздачі.

4. Розробка версії веб-додатку для смартфонів.

5. Впровадження додаткових заходів безпеки для захисту конфіденційної інформації користувачів.

Ці кроки дозволять піднятися на новий рівень функціональності та ефективності телеграм-боту, роблячи його ще більш привабливим для широкого кола користувачів.

3.4 Аналіз отриманих результатів

Розділ систематизує основні висновки та ідентифікує патерни, виявлені у нашому аналізі. Цей розділ вирішує завдання виявлення асоціацій та групування даних, розкриваючи цікаві взаємозв'язки та тенденції. Детальний аналіз донорів крові включає проведення кластерного аналізу для групування схожих донорів та виявлення спільних характеристик.

У розділі подаються результати аналізу асоціацій, де визначаються часті комбінації ознак серед донорів крові. Аналіз асоціацій допомагає розкрити взаємозв'язки між різними характеристиками, що може бути корисним для оптимізації процесу здачі крові та управління запасами.

Додатково, розділ включає прогнозування на основі отриманих даних. Використовуючи аналітичні методи, ми розглядаємо можливості прогнозування змін та тенденцій у донорській базі, що дозволяє вчасно адаптувати стратегії управління. Ця інформація виробляє ключові висновки та рекомендації, щоб

покращити ефективність системи управління донорами крові та забезпечити сталість постачання.

На рисунку 3.7 маємо такий сформований json файл з даними донорів.

```

1  [
2  {
3    "last_name": "Джуба",
4    "first_name": "Авелій",
5    "date_of_birth": "1980-05-23",
6    "sex": "чоловіча",
7    "phone_number": "+380 89 448-47-26",
8    "height": 153.55,
9    "weight": 65.1,
10   "blood_group": 4,
11   "city": "хутір Надвірна",
12   "last_donation_date": "2023-02-17"
13  },
14  {
15   "last_name": "Фартушняк",
16   "first_name": "Оксана",
17   "date_of_birth": "1960-05-16",
18   "sex": "жіноча",
19   "phone_number": "070 118-77-01",
20   "height": 151.8,
21   "weight": 84.44,
22   "blood_group": 1,
23   "city": "місто Новгород-Сіверський",
24   "last_donation_date": "2023-04-26"
25  },
26  {
27   "last_name": "Девдок",
28   "first_name": "Богуслава",
29   "date_of_birth": "1986-10-11",
30   "sex": "жіноча",
31   "phone_number": "+380 79 269-63-10",
32   "height": 183.24,
33   "weight": 52.75,
34   "blood_group": -1,
35   "city": "село Святогірськ",
36   "last_donation_date": "2023-06-28"
37  },

```

Рисунок 3.7 – json файл з даними донорів

Проведемо кластерний аналіз (рис. 3.8).

```

1  import json
2  from sklearn.cluster import KMeans
3  from sklearn.preprocessing import LabelEncoder
4  from datetime import datetime
5
6  # Зчитування даних з файлу
7  with open('dataset.json', 'r', encoding='utf-8') as file:
8      dataset = json.load(file)
9
10 # Перетворення категоріальних даних (тип крові)
11 label_encoder = LabelEncoder()
12 blood_group_encoded = label_encoder.fit_transform([person['blood_group'] for person in dataset])
13
14 # Перетворення дати в числовий формат (кількість днів від певної початкової дати)
15 start_date = datetime.strptime('2023-01-01', '%Y-%m-%d')
16 days_since_start = [(datetime.strptime(person['last_donation_date'], '%Y-%m-%d') - start_date).days for person in dataset]
17
18 # Вибір полів для кластеризації (дата останньої здачі крові та тип крові)
19 features = list(zip(days_since_start, blood_group_encoded))
20
21 # Кластерний аналіз
22 kmeans = KMeans(n_clusters=3, random_state=42)
23 kmeans.fit(features)
24
25 # Виведення результатів кластеризації
26 for i, person in enumerate(dataset):
27     cluster_label = kmeans.labels_[i]
28     print(f"{person['first_name']} {person['last_name']} - Cluster {cluster_label}")
29

```

Рисунок 3.8 – Код для проведення кластерного аналізу

У цьому коді виконується кластерний аналіз даних про донорів крові з використанням бібліотеки Scikit-learn на мові програмування Python. Етапи:

1. Зчитування даних. Інформація про донорів крові зчитується з файлу dataset.json.
2. Перетворення категоріальних даних. Тип крові перетворюється з текстового формату до числового за допомогою LabelEncoder, що дозволяє алгоритму працювати з цими даними.
3. Перетворення дати. Дата останньої здачі крові перетворюється в числовий формат, представляючи кількість днів від певної початкової дати.
4. Вибір полів для кластеризації. Визначаються ознаки, які будуть використовуватися для кластеризації – це дата останньої здачі крові та тип крові.
5. Кластерний аналіз. Застосовується алгоритм K-Means для розділення донорів на групи (кластери) на основі обраних ознак: дата останньої здачі крові та група крові.

Алгоритм K-Means - це метод кластеризації, який призначений для групування схожих об'єктів у кластери. Він базується на мінімізації середньоквадратичного відхилення в кластерах, що робить його ефективним для роботи з числовими даними. Нижче наведено опис алгоритму K-Means:

1. Вибір кількості кластерів (K). Кількість кластерів, яку ви хочете отримати, визначається перед виконанням алгоритму. В даному випадку отримуємо 3 кластери.
2. Ініціалізація центрів кластерів. Випадковим чином вибираються K точок з набору даних, і вони використовуються як початкові центри кластерів.
3. Призначення точок до кластерів. Кожна точка даних призначається тому кластеру, центр якого є найближчим до неї. Використовується метрика відстані, така як евклідова відстань.
4. Перерахунок центрів кластерів. Для кожного кластера обчислюється новий центр, який представляє середнє арифметичне всіх точок, належних до цього кластера.

5. Повторення кроків 3-4. Кроки 3 і 4 повторюються до тих пір, поки центри кластерів не стабілізуються або досягнута максимальна кількість ітерацій.

6. Коли алгоритм збігається, кожна точка даних віднесена до одного з кластерів, і центри кластерів зафіксовані.

Алгоритм K-Means має деякі недоліки, такі як чутливість до початкового вибору центрів та вплив викидів на результат. Також, він не ефективний для даних з складною структурою. Важливо провести попередню обробку даних та вибрати оптимальне значення K для досягнення кращих результатів.

6. Виведення результатів. Виводяться результати кластеризації, де кожному донору присвоюється мітка кластеру, і виводяться його ім'я та прізвище разом із міткою кластеру.

Отже, код визначає групи подібних донорів на основі часу від дати останньої здачі крові та типу крові. Розглянемо результат на рисунку 3.9.

```

Аврелій Дзюба - Cluster 0
Оксана Фартушняк - Cluster 2
Богуслава Девдюк - Cluster 2
Святослав Юрчишин - Cluster 0
Приска Габелко - Cluster 0
Борис Забарний - Cluster 2
Роксолана Василенко - Cluster 2
Валерій Рудик - Cluster 2
Анжела Яремко - Cluster 1
Алла Давиденко - Cluster 0
Опанас Удовиченко - Cluster 0
Віолетта Семенченко - Cluster 1
Михайлина Бараник - Cluster 2
Володимира Чаленко - Cluster 2
Ірена Акименко - Cluster 1
Софія Рябенко - Cluster 1
Хома Іванченко - Cluster 0
Сергій Баштан - Cluster 1
Мілена Приходько - Cluster 1
Левко Скопенко - Cluster 0
Федір Тичина - Cluster 1
Сергій Гаврилишин - Cluster 2
Ярина Дубенко - Cluster 1
Гліб Артим - Cluster 1
Артем Коноplenко - Cluster 1
Венедикт Батіг - Cluster 2
Гордій Аврамчук - Cluster 2
Юстим Байрак - Cluster 1
Григорій Тимченко - Cluster 0

```

Рисунок 3.9 – Результат кластеризації

Результати кластеризації вказують на групування об'єктів на основі їхніх властивостей. Дані були кластеризовані за трьома кластерами (Cluster 0, Cluster

1, Cluster 2). Проведемо детальний аналіз отриманих даних для кожного кластера.

Кластер №0 на рисунку 3.10 містить осіб із різними типами крові (4, 0, -1, 2). Також особи у цьому кластері мають різні розподіли дат останньої здачі крові.

Cluster 0:

- Богуслава Девдюк
- Пріска Габелко
- Борис Забарний
- Володимира Чаленко
- Хома Іванченко
- Мілена Приходько
- Сергій Гаврилишин
- Єлисей Черненко
- Гліб Артим
- Венедикт Батіг
- Єва Скоробогатько
- Демид Лукаш
- Григорій Захаренко

Рисунок 3.10 – Кластер №0

Кластер №1 на рисунку 3.11 має осіб із типами крові 1, -4, 3, 2. Також особи у цьому кластері мають спільну ознаку в тому, що їхня дата останньої здачі крові розташована наближено до 2023-06-28.

Cluster 1:

- Роксолана Василенко
- Опанас Удовиченко
- Ірена Акименко
- Софія Рябченко
- Левко Скопенко
- Федір Тичина
- Віра Матяш
- Ярина Дубенко
- Сергій Баштан
- Юстим Байрак
- Григорій Тимченко
- Марина Чаленко
- Лілія Вишиваний
- Єлисей Дубина
- Єва Скоробогатько
- Охрім Оберемко

Рисунок 3.11 – Кластер №1

Кластер №2 на рисунку 3.12 має осіб із типами крові 4, 2, 1, -2. Також особи у цьому кластері мають різні розподіли дат останньої здачі крові.

Cluster 2:

- Аврелій Дзюба
- Оксана Фартушняк
- Святослав Юрчишин
- Валерій Рудик
- Алла Давиденко
- Віолетта Семенченко
- Михайлина Бараник
- Сергій Гаврилишин
- Анжела Яремко
- Левко Скопенко
- Артем Коноplenко
- Гордій Аврамчук
- Єлисей Черненко
- Ігнат Балабан
- Семен Фоменко
- Віктор Яковенко
- Григорій Захаренко
- Віолетта Чуйко
- Розалія Іващенко
- Азар Непорожній

Рисунок 3.12 – Кластер №2

Далі проведемо асоціативний аналіз (рис. 3.13).

```

1  import json
2  import pandas as pd
3
4  # Зчитуємо дані з JSON-файлу
5  with open('dataset.json', 'r', encoding='utf-8') as file:
6      data = json.load(file)
7
8  # Створюємо DataFrame з даних
9  df = pd.DataFrame(data)
10
11 # Виводимо асоціативний аналіз
12 associative_analysis = df.describe(include='all')
13 print(associative_analysis)
14

```

Рисунок 3.13 – Код для проведення асоціативного аналізу

Цей код використовує бібліотеку pandas для обробки та аналізу даних, що містять інформацію про донорів крові, збережену у форматі JSON. Основні кроки:

1. Зчитування даних з файлу. Відбувається завантаження даних з файлу `dataset.json`.
2. Створення `DataFrame`. `Pandas` створює `DataFrame`, який представляє собою табличну структуру для зручного аналізу даних. Кожний рядок цього `DataFrame` відповідає окремому донорові, а колонки містять різні атрибути (ім'я, вік, стать тощо).
3. Асоціативний аналіз. Використовується метод `describe()` для створення асоціативного аналізу, який включає статистичну інформацію про всі колонки `DataFrame`. Це може включати середнє значення, стандартне відхилення, мінімальне та максимальне значення, а також різні кількісні та якісні параметри для кожного атрибута.
4. Виведення результатів. Аналіз виводиться на екран для подальшого огляду та розуміння основних характеристик даних.

Отже, код допомагає візуалізувати статистичну інформацію про дані та здійснює асоціативний аналіз для кожного атрибута в `DataFrame`. Розглянемо результат на рисунку 3.14.

```

>>> %Run script.py
      last_name first_name ...          city last_donation_date
count          50          50 ...           50           50
unique          48          45 ...           50           47
top      Чаленко      Віра ...  хутір Надвірна      2022-12-24
freq           2           2 ...             1             2
mean           NaN          NaN ...           NaN           NaN
std            NaN          NaN ...           NaN           NaN
min            NaN          NaN ...           NaN           NaN
25%            NaN          NaN ...           NaN           NaN
50%            NaN          NaN ...           NaN           NaN
75%            NaN          NaN ...           NaN           NaN
max            NaN          NaN ...           NaN           NaN

[11 rows x 10 columns]
>>>

```

Рисунок 3.14 – Результат асоціативного аналізу

В результаті отримали вивід, який надає функція `describe()` бібліотеки `Pandas`. У цьому виводі відображається статистичний аналіз для кожного стовпця у наборі даних. Деяка важлива інформація, яку можна отримати з цього аналізу, включає кількість непорожніх значень (`count`), кількість унікальних значень

(unique), найчастіше значення, яке зустрічається (top), і кількість разів, коли найчастіше зустрічається це значення (freq).

Для числових стовпців (таких як «height» та «weight») виводяться статистичні параметри, такі як середнє значення (mean), стандартне відхилення (std), мінімальне (min) та максимальне (max) значення, а також квартилі. Давайте розглянемо деякі деталі асоціативного аналізу:

1. count (кількість). Кожен стовпець має 50 значень, що означає, що у нас є повний набір даних для 50 осіб.

2. unique (унікальні значення). Деякі стовпці, такі як «last_name» та «first_name», мають менше унікальних значень, що може бути пов'язано з повторенням прізвищ та імен.

3. «blood_group» має 4 унікальних значення.

4. top (найчастіше значення, що зустрічається). Поле «city» має значення «хутір Надвірна» - значення, що найчастіше зустрічається.

5. mean (середнє значення). Деякі стовпці, такі як «height» та «weight», мають середні значення, що може бути корисно при подальшому аналізі.

6. std (стандартне відхилення). «height» та «weight» мають стандартне відхилення, що вказує на розкид значень навколо середнього.

7. min та max (мінімальне та максимальне значення). «date_of_birth» показує, що ви маєте діапазон дат народження від 1960-05-16 до 1986-10-11. «height» та «weight» показують діапазон від мінімального до максимального значення.

Далі проведемо прогнозування даних (рис. 3.15).


```

1 import json
2 from datetime import datetime, timedelta
3
4 def calculate_possible_donors(dataset, target_date):
5     # Функція для розрахунку кількості можливих донорів на конкретну дату
6     # Кількість днів між кроводачами
7     min_days_between_donations = 60
8
9     # Перетворення строки в дату
10    target_date = datetime.strptime(target_date, '%Y-%m-%d')
11
12    # Лічильники для підрахунку кількості донорів та груп крові
13    possible_donors_count = 0
14    blood_type_count = {"O+": 0, "O-": 0, "A+": 0, "A-": 0, "B+": 0, "B-": 0, "AB+": 0, "AB-": 0}
15
16    # Проходження по кожному запису в датасеті
17    for donor in dataset:
18        # Перетворення строки в дату для останнього проходу
19        last_donation_date = datetime.strptime(donor['last_donation_date'], '%Y-%m-%d')
20        # Розрахунок різниці в днях між останнім проходом та цільовою датою
21        days_since_last_donation = (target_date - last_donation_date).days
22        # Якщо пройшло більше 60 днів, додати донора до можливих донорів
23        if days_since_last_donation >= min_days_between_donations:
24            possible_donors_count += 1
25            # Збільшення лічильника для відповідної групи крові
26            blood_type = donor['blood_type'] + donor['rhesus_factor']
27            if blood_type in blood_type_count:
28                blood_type_count[blood_type] += 1
29    return possible_donors_count, blood_type_count
30

```

Рисунок 3.15 – Код для проведення прогнозування

Виконується обчислення можливої кількості донорів на задану дату та рахується кількість донорів для кожної групи крові. Давайте розглянемо його детальніше:

Функція `calculate_possible_donors` розраховує кількість можливих донорів та лічильник для кожної групи крові на задану дату. Вона приймає два параметри: `dataset` - дані про донорів у форматі JSON та `target_date` - цільова дата, для якої робиться розрахунок.

Далі іде розрахунок кількості днів між кроводачами. Визначається мінімальна кількість днів (`min_days_between_donations`), яку повинен пройти донор між кроводачами. Цільова дата перетворюється з строки в об'єкт дати для подальших розрахунків. Цикл проглядає кожного донора у датасеті. Обчислюється різниця в днях між цільовою датою та останнім проходженням донором.

Якщо пройшло більше днів, ніж `min_days_between_donations`, донор додається до можливих донорів, і лічильники для відповідних груп крові збільшуються.

Після введення цільової дати користувачем результати обчислень виводяться для надання інформації про те, скільки людей може прийти та кількість донорів для кожної групи крові (рис. 3.16).

```

31 # Зчитування датасету з файлу
32 with open('dataset.json', 'r', encoding='utf-8') as file:
33     dataset = json.load(file)
34
35 # Введення цільової дати
36 target_date = input("Введіть цільову дату у форматі YYYY-MM-DD: ")
37
38 # Розрахунок можливих донорів та груп крові
39 result_count, result_blood_types = calculate_possible_donors(dataset, target_date)
40
41 # Виведення результатів
42 print(f"На {target_date} може прийти здати кров {result_count} людей.")
43
44 # Виведення кількості донорів з різними групами крові
45 print("Кількість донорів з різними групами крові:")
46 for blood_type, count in result_blood_types.items():
47     print(f"{blood_type}: {count} людей.")
48

```

Рисунок 3.16 – Зчитування та введення необхідних даних

Отримані результати виводяться на екран, включаючи кількість можливих донорів та кількість донорів для кожної групи крові (рис. 3.17).

<p>Введіть цільову дату у форматі YYYY-MM-DD: 2023-02-01</p> <p>На 2023-02-01 може прийти здати кров 1 людей.</p> <p>Кількість донорів з різними групами крові:</p> <p>O+: 0 людей.</p> <p>O-: 0 людей.</p> <p>A+: 0 людей.</p> <p>A-: 0 людей.</p> <p>B+: 0 людей.</p> <p>B-: 0 людей.</p> <p>AB+: 1 людей.</p> <p>AB-: 0 людей.</p>	<p>Введіть цільову дату у форматі YYYY-MM-DD: 2023-12-12</p> <p>На 2023-12-12 може прийти здати кров 46 людей.</p> <p>Кількість донорів з різними групами крові:</p> <p>O+: 7 людей.</p> <p>O-: 6 людей.</p> <p>A+: 9 людей.</p> <p>A-: 5 людей.</p> <p>B+: 3 людей.</p> <p>B-: 5 людей.</p> <p>AB+: 7 людей.</p> <p>AB-: 4 людей.</p>
---	--

Рисунок 3.17 – Результати прогнозування

Отримані результати аналізу мають важливе значення для управління резервами крові та організації процесу здачі крові. Отримана кількість можливих донорів на певну дату дозволяє зрозуміти, скільки людей може прийти та взяти участь у здачі крові в цей конкретний день. Це допомагає організаторам кроводачі оцінити загальну масштабність та приготуватися до прийому необхідної кількості донорів.

Розподіл кількості можливих донорів за групами крові надає цінну інформацію для управління резервами. Знання, скільки людей кожної групи

крові може прийти, дозволяє ефективніше планувати та розподіляти резерви крові. Наприклад, якщо виявляється, що певні групи крові менш представлені, можна спрямувати додаткові зусилля на привертання донорів цих груп.

Загальний висновок полягає в тому, що на певну цільову дату можливо очікувати певну кількість донорів, і розподіл за групами крові дозволяє ефективно керувати запасами та забезпечити достатню кількість крові для різних груп отримувачів.



Висновок до розділу 3

У цьому розділі була представлена структура інтелектуальної інформаційно-аналітичної системи донорства крові. Ця структура є ключовим елементом для успішної розробки та функціонування системи.

Структура системи включає такі компоненти, як вхідні дані, обробку та аналіз даних, базу даних, інтелектуальні алгоритми та моделі, веб-інтерфейс, сповіщення та комунікацію. Ці компоненти допомагають забезпечити зручний доступ до інформації про донорів та медичні записи, ефективну обробку та аналіз даних, а також забезпечують зручну взаємодію між користувачами та системою.

Структура системи також сприяє автоматизації багатьох процесів, таких як запис на збір крові, нагадування про планові збори, надсилання результатів аналізів та інші функції. Крім того, наявність інтелектуальних алгоритмів та моделей дозволяє системі робити прогнози, рекомендації та аналізувати дані для прийняття обґрунтованих рішень.

Ця структура є гнучкою та може адаптуватися до конкретних потреб та вимог системи донорства крові. Вона дозволяє забезпечити ефективне функціонування системи, зручну взаємодію з користувачами та поліпшення якості та ефективності донорських програм.

ВИСНОВКИ

У результаті проведення дослідження та аналізу була розроблена інтелектуальна інформаційно-аналітична система донорства крові. Ця система має великий потенціал у поліпшенні організації та управління донорськими програмами, забезпечуючи ефективність, швидкість та точність обробки даних та роботу з інформацією.

Було поставлено та виконано такі завдання:

1. Проаналізовано аналоги інтелектуальних інформаційно-аналітичних систем донорства крові. Висвітлені їх переваги та недоліки.
2. Досліджено методи для збору та аналізу даних, а також основні складові системи донорства крові, включаючи реєстрацію донорів, управління запасами крові, організацію донорських сеансів, сповіщення та нагадування донорам.
3. Побудовано модель інтелектуальної інформаційно – аналітичної системи донорства крові для міста Вінниці.
4. Проаналізовано та обрано інструменти для реалізації інтелектуальної інформаційно – аналітичної системи донорства крові.
5. Реалізовано інтелектуальну інформаційно – аналітичну систему донорства крові для міста Вінниці.
6. Сформовано базу даних донорів і проведено аналіз для оцінки ситуації.

Основними перевагами інтелектуальної інформаційно-аналітичної системи донорства крові є:

1. Зручний та швидкий доступ до інформації. Система дозволяє легко знаходити та отримувати доступ до даних про донорів, медичні записи, запаси крові та іншу важливу інформацію;
2. Автоматизація процесів. Система автоматизує багато процесів, включаючи запис на збір крові, нагадування та сповіщення про важливі події. Це допомагає зменшити рутинну роботу та покращити ефективність роботи.

3. Інтелектуальний аналіз та прогнозування. Застосування інтелектуальних алгоритмів та моделей дозволяє системі аналізувати дані, робити прогнози та рекомендації. Це допомагає виявляти потенційних донорів, прогнозувати попит на кров та планувати збори більш ефективно.

4. Підвищення безпеки та точності. Система допомагає забезпечити високий рівень безпеки та точності обробки даних. Вона дозволяє зберігати медичну інформацію та персональні дані донорів в безпечному середовищі і захищати їх від несанкціонованого доступу.

5. Поліпшення взаємодії з користувачами. Інтелектуальна інформаційно-аналітична система донорства крові забезпечує зручний та користувачам зрозумілий інтерфейс, що полегшує їх взаємодію з системою. Це допомагає залучити більше донорів та поліпшити загальний досвід користувачів.

Загальною метою інтелектуальної інформаційно-аналітичної системи донорства крові було покращення якості та ефективності донорських програм, забезпечення швидкого та зручного доступу до інформації, автоматизація процесів та надання інтелектуального аналізу для прийняття обґрунтованих рішень.

В результаті впровадження інтелектуальної інформаційно-аналітичної системи донорства крові можливе поліпшення організації та управління донорськими програмами, забезпечення швидкої та точної обробки даних, поліпшення безпеки та зручності взаємодії з користувачами, а також підвищення ефективності та результативності донорських програм в цілому.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Зелінська О.В., Гавліцький В.Ф., Якубич К.О. Проектування інтелектуальної інформаційно-аналітичної системи донорства крові. Електронний журнал «Наука і техніка сьогодні». Випуск №13(27). Київ. 2023. С. 748 – 758.
2. Зелінська О.В., Потапова Н.А., Волонтир Л.О. Інформаційні системи та технології в галузі : навчальний посібник. Вінниця. ВНАУ. 2020. 263 с.
3. Якубич К.О., Зелінська О.В. Інформаційно-аналітична система донорства крові: матеріали IV Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Прикладні інформаційні технології». Вінниця. ДонНУ імені Василя Стуса. 2023. С. 202-204.
4. Суханов А.О., Зелінська О.В. Використання сучасних інтелектуальних інформаційних технології. Комп'ютерні технології обробки даних: матеріали II Всеукраїнської науково-практичної конференції. Вінниця. ДонНУ імені Василя Стуса. 2021. С. 154-155.
5. Якубич К.О. Використання математичних моделей для прогнозування та оптимізації роботи інформаційних систем: матеріали IV Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Прикладні інформаційні технології». Вінниця. ДонНУ імені Василя Стуса. 2023. С. 321-322.
6. Постанова від 17 лютого 2023 р. №143 Київ «Про створення інформаційно-комунікаційного комплексу системи кров» [Електронний ресурс] URL: <https://zakon.rada.gov.ua/laws/show/143-2023-%D0%BF#n12>. (дата звернення 15.10.23 р.).
7. ДонорUA: Всеукраїнська система рекрутингу донорів крові [Електронний ресурс] URL: <https://www.donor.ua/> (дата звернення 10.10.23 р.).
8. Kathleen Chell, Tanya E Davison, Barbara Masser, Kyle Jensen. A systematic review of incentives in blood donation. Transfusion 58 (1). 2018. 242-254.

9. Yong-Hua Yin, Chang-Qing Li, Zhong Liu. Blood donation in China: sustaining efforts and challenges in achieving safety and availability. *Transfusion* 55 (10). 2015. 2523-2530.
10. A Ali, Israt Jahan, A Islam, S Parvez. Blood donation management system. *American Journal of Engineering Research* 4 (6). 2015. 123-136.
11. Seda Baş, Giuliana Carello, Ettore Lanzarone, Zeynep Ocak, Semih Yalçındağ. Management of blood donation system: literature review and research perspectives. *Health Care Systems Engineering for Scientists and Practitioners: HCSE*. France. 2016. 121-132.
12. Славінська І. 2/3 пацієнтів в Україні не отримують кров або взагалі, або вчасно. *Суспільне новини*. 2023. [Електронний ресурс] URL: <https://suspilne.media/507250-23-pacientiv-v-ukraini-ne-otrimuut-krov-abo-vzagali-abo-vcasno-spivzasnovnica-donoua/>.
13. Донорство крові має бути систематичним, а не ситуативним – координаторка ДонорUA. *Українформ - мультимедійна платформа іномовлення України*. 2023. [Електронний ресурс] URL: <https://www.ukrinform.ua/rubric-health/3755717-donorstvo-krovi-mae-buti-sistematicnim-a-ne-situativnim-koordinator-donoua.html> (дата звернення 20.09.23 р.).
14. Богданьок. О. Центрам служби крові потрібні донори крові резус-негативної належності — МОЗ. *Суспільне новини*. 2023. [Електронний ресурс] URL: <https://suspilne.media/435918-centram-sluzbi-krovi-potribni-donori-krovi-rezus-negativnoi-naleznosti-moz/> (дата звернення 15.09.23 р.).
15. Almetwally M Mostafa, Ahmed E Youssef, Gamal Alshorbagy. A framework for a smart social blood donation system based on mobile cloud computing. *arXiv preprint arXiv:1412.2014.7276*.
16. Lalmohan Dutta, Giridhar Maji, Partha Ghosh, Soumya Sen. An integrated blood donation campaign management system. *Contemporary Advances in Innovative and Applicable Information Technology: Proceedings of ICCAIAIT*. 2019. 133-143.

17. Shadi AlZu'bi, Darah Aqel, Ala Mughaid. Recent intelligent approaches for managing and optimizing smart blood donation process. 2021 International Conference on Information Technology (ICIT). 2021. 679-684.
18. Литвин В.В., Пасічник В.В., Шаховська Н.Б. Проектування інформаційних систем: навчальний посібник. 2021. 380 с.
19. Чіома Е.В., Зелінська О.В. Інформаційна система управління особистими фінансами. Прикладні аспекти сучасних міждисциплінарних досліджень: матеріали I Всеукраїнської науково-практичної конференції. Вінниця: ДонНУ імені Василя Стуса. 2021. С. 121-123.
20. Марченко А.В. Проектування інформаційних систем : навчальний посібник. 2015. 40 с.
21. Литвин В.В., Пасічник В.В., Шаховська Н.Б. Проектування інформаційних систем: навчальний посібник. 2021. 380 с.
22. Stephen Cass. The top programming languages: Our latest rankings put Python on top-again-[Careers]. IEEE Spectrum 57 (8). 2020. 22-22.
23. Nurul Ashikin Abd Samad, Zalihar Embong, Maizatul Akhmar Mohamad Nor. Development of Telegram Bot in Supporting the Learning Process of Object-Oriented Programming. LIS International Conference 8 (September). 2022. 134-144.
24. Rianto Rianto, Alam Rahmatulloh, Teguh Anugrah Firmansah. Telegram bot implementation in academic information services with the forward chaining method. Sinkron: jurnal dan penelitian teknik informatika 3 (2). 2019. 73-78.
25. Telegram Bot API [Електронний ресурс] URL: <https://core.telegram.org/bots/api> (дата звернення 17.08.23 р.).
26. Bots: An introduction for developers. [Електронний ресурс] URL: <https://core.telegram.org/bots> (дата звернення 11.08.23 р.).
27. Documentation Node.js Telegram Bot API [Електронний ресурс] URL: <https://www.npmjs.com/package/node-telegram-bot-api> (дата звернення 28.08.23 р.).
28. Ning Zhong, Jiming Liu. Intelligent technologies for information analysis. Springer Science & Business Media. 2013.

29. Prateek Rawat, Archana N Mahajan. ReactJS: A modern web development framework. *International Journal of Innovative Science and Research Technology* 5 (11). 2020. C. 698-702.
30. Sanchit Aggarwal. Modern web-development using reactjs. *International Journal of Recent Research Aspects* 5 (1). 2018. C. 133-137.
31. Archana Bhalla, Shivangi Garg, Priyangi Singh. Present day web-development using reactjs. *International Research Journal of Engineering and Technology* 7 (05). 2020.
32. Cory Gackenheimer. *Introduction to React*. Apress. 2015. 129c.
33. Kwangwon Sun, Sukyoung Ryu. Analysis of JavaScript programs: Challenges and research trends. *ACM Computing Surveys (CSUR)* 50 (4). 2017. C.1-34.
34. Addy Osmani. *Learning JavaScript design patterns*. O'Reilly Media, Inc. 2023. 296 c.
35. Shannon Bradshaw, Eoin Brazil, Kristina Chodorow. *MongoDB: the definitive guide: powerful and scalable data storage*. O'Reilly Media. 2019. 511c.
36. Gansen Zhao, Weichai Huang, Shunlin Liang, Yong Tang. Modeling MongoDB with relational model. *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*. 2013. C.115-121.
37. Mithun Satheesh, Bruno Joseph D'mello, Jason Krol. *Web development with MongoDB and NodeJs*. Packt Publishing Ltd. 2015. 300 c.
38. Anuradha Kanade, Arpita Gopal, Shantanu Kanade. A study of normalization and embedding in MongoDB. *2014 IEEE International Advance Computing Conference (IACC)*. 2014. C.416-421.
39. Shakuntala Gupta Edward, Navin Sabharwal. *Practical MongoDB: Architecting, Developing, and Administering MongoDB*. Apress. 2015. 263c.
40. Jan L Harrington. *Relational database design and implementation*. Morgan Kaufmann. 2016. 689 c.
41. Adrienne Watt, Nelson Eng. *Database design*. BCcampus. 2014. 144 c.

42. Felipe Pezoa, Juan L Reutter, Fernando Suarez, Martín Ugarte, Domagoj Vrgoč. Foundations of JSON schema. Proceedings of the 25th international conference on World Wide Web. 2016. С. 263-273.
43. Lindsay Bassett. Introduction to JavaScript object notation: a to-the-point guide to JSON. O'Reilly Media, Inc. 2015. 126 с.
44. Коваленко О. С., Добровська Л. М. Проектування інформаційних систем: Загальні питання теорії проектування ІС. Київ : КПІ ім. Ігоря Сікорського. 2020. 192 с.
45. Карпенко М. Ю., Манакова Н. О., Гавриленко І. О. Технології створення програмних продуктів та інформаційних систем. Харків : ХНУМГ ім. О. М. Бекетова. 2017. 93 с.
46. Недашківський О.Л. Планування та проектування інформаційних систем. Київ. 2014. 215 с.
47. Ременяк Л.В. Проектування інформаційних систем: конспект лекцій. Одеса: Одеський державний екологічний університет. 2016. 152 с.
48. Ланде Д.В., Субач І.Ю., Бояринова Ю.Є. Основи теорії і практики інтелектуального аналізу даних у сфері кібербезпеки: навчальний посібник. Київ: КПІ ім. Ігоря Сікорського. 2018. 300 с.
49. Рогушина Ю.В. Засоби та методи аналізу неструктурованих даних. Проблеми програмування. 2019. С. 57-77.
50. Бідюк П. І. Ймовірно-статистичні методи моделювання і прогнозування : монографія. Миколаїв : Вид-во ЧДУ ім. Петра Могили. 2014. 440 с.
51. Гороховатський В. О. Методи інтелектуального аналізу та оброблення даних : навч. посіб. Харків : ХНУРЕ. 2021. 92 с.
52. Roxy Peck, Tom Short, Chris Olsen. Introduction to statistics and data analysis. Cengage Learning. 2020. 896 с.

Додаток 2 до наказу
від «31» березня 2023 року
№119/05

ДЕКЛАРАЦІЯ

про дотримання академічної доброчесності

Я, _____

Повністю вказується ПІБ та статус (посада для працівників, освітня (освітньо-наукова) програма – для здобувачів вищої освіти)

що нижче підписалась/підписався, розуміючи та підтримуючи загально визнані засади справедливості, доброчесності та законності,

ЗОБОВ'ЯЗУЮСЬ:

дотримуватися принципів та правил академічної доброчесності, що визначені законодавством України, локальними нормативними актами Донецького національного університету імені Василя Стуса, положеннями, правилами, умовами, визначеними іншими суб'єктами, та не допускати їх порушення.

ПІДТВЕРДЖУЮ:

що мені відомі положення статті 42 Закону України «Про освіту»;
що у даній роботі не представляла/представляв чийсь роботи повністю або частково як свої власні. Там, де я скористалася/скористався працею інших, я зробила/зробив відповідні посилання на джерела інформації;

що дана робота не передавалась іншим особам і подається вперше, не порушує авторських та суміжних прав закріплених статтями 21-25 Закону України «Про авторське право та суміжні права», а дані та інформація не отримувались в недозволеній спосіб.

УСВІДОМЛЮЮ:

що ця робота може бути перевірена університетом на плагіат або інші порушення академічної доброчесності, в тому числі з використанням спеціалізованих сервісів;

що у разі порушення академічної доброчесності, до мене можуть бути застосовані процедури, передбачені законодавством України та Кодексом академічної доброчесності та корпоративної етики Донецького національного університету імені Василя Стуса, іншими локальними нормативними актами університету, та я можу бути притягнута/притягнутий до академічної відповідальності.

_____ (дата)

_____ (підпис)