

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ЗАБЛУДА МИКОЛА ОЛЕКСАНДРОВИЧ

Допускається до захисту:

В.о. завідувач кафедри ПМ,
_____ Трофименко О.Д.

« _____ » _____ 20__ р.

СИСТЕМА ОРГАНІЗАЦІЇ ІНТЕРАКТИВНОГО ГОЛОСУВАННЯ

Спеціальність 113 Прикладна математика

Кваліфікаційна (бакалаврська) робота

Керівник:

Ветров О. С., старший викладач
кафедри Прикладної математики

Оцінка ____ / ____ / _____

Голова ЕК: _____

АНОТАЦІЯ

Заблуда М. О. Система організації інтерактивного голосування. Спеціальність 113 «Прикладна математика». Донецький національний університет імені Василя Стуса, Вінниця, 2021.

У кваліфікаційній (бакалаврській) роботі досліджено системи електронного голосування, які використовуються сьогодні у різних сферах життя людей, а також ті, які активно розвиваються та набувають поширення у зв'язку зі стрімким розвитком технологій та способом життя людей. Розглянуто методи голосування в групах, на основі яких можна створювати програмне забезпечення для інтерактивного голосування в сфері освіти для студентів вищих навчальних закладів та інтегрувати їх у лекційні та практичні заняття.

Об'єкт дослідження – системи інтерактивного голосування.

Предмет дослідження – системи інтерактивного голосування для миттєвої комунікації між студентами та викладачами для покращення якості викладання матеріалу лектором.

Мета роботи полягає у розробці інтернет ресурсу для запровадження системи організації інтерактивного голосування серед студентів задля покращення якості викладання матеріалу викладачами та миттєвої взаємодії між студентом та лектором.

Результатом дослідження проведеного у бакалаврській кваліфікаційній роботі було реалізовано програмне забезпечення для інтерактивного голосування для покращення наповнення та якості матеріалу, який надається на лекційних заняттях.

Ключові слова: електронні системи голосування, парадокс Кондорсе, парадокс Борда, консенсус, схвальне голосування, інтерактивна освіта.

57 с., 2 табл., 11 рис., 3 дод., 24 джерела.

ABSTRACT

Zabluda M. Interactive voting system. Specialty 113 «Applied Mathematics». Vasyl' Stus Donetsk National University, Vinnytsia, 2021.

The qualification (bachelor's) work examines the electronic voting systems used today in various spheres of human life, as well as those that are actively developing and gaining popularity due to the rapid development of technology and lifestyle. The methods of voting in groups are considered, on the basis of which it is possible to create software for interactive voting in the field of education for students of higher educational institutions and to integrate them into lectures and practical classes.

Study object – interactive voting systems.

Scope of research is interactive voting systems for instant communication between students and teachers to improve the quality of teaching material by the lecturer.

Goal of research is to develop an Internet resource for the introduction of a system of interactive voting among students in order to improve the quality of teaching material by teachers and instant interaction between the student and the lecturer.

The result of the research conducted in the bachelor's thesis was the creation of the software for interactive voting was implemented to improve the content and quality of the material provided in the lectures.

Keywords: Electronic Voting Systems, Condorcet's Paradox, Board's Paradox, consensus, approval voting, interactive education.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ТА МЕТОДИ СИСТЕМ ГОЛОСУВАННЯ В ГРУПАХ .	12
1.1. Теорія голосування. Різновиди систем голосування	12
1.2. Методи систем голосування	14
1.2.1. Метод голосування відносної більшості.....	16
1.2.2. Метод Кондорсе.....	17
1.2.3. Метод Борда (парадокс Борда)	17
1.2.4. Схвальне голосування.....	19
1.2.5. Методи пошуку консенсусу.....	20
1.3. Переваги та недоліки голосування проти досягнення консенсусу	20
1.4. Типологія електронних систем голосування	22
РОЗДІЛ 2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЕЛЕКТРОННОГО ГОЛОСУВАННЯ.....	25
2.1. Вибір мови програмування.....	25
2.2. Інтегровані середовища розробки.....	26
2.3. Переваги використання Angular	28
2.4. Архітектура Angular-проекту	29
2.5. TypeScript в Angular-проекті.....	32
2.6. Структура робочої області та модулі JavaScript	32
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОГО ГОЛОСУВАННЯ В ГРУПАХ ДЛЯ ЕФЕКТИВНОЇ КОМУНІКАЦІЇ НА ЛЕКЦІЙНИХ ЗАНЯТТЯХ У ВИЩИХ НАВЧАЛЬНИХ ЗАЛАДАХ.....	35
3.1. Ініціалізація проекту	35
3.2. UML-діаграми програмного продукту.....	36
3.3. Реалізація функціоналу для ефективної комунікації на лекційних заняттях	37
3.4. Реалізація функціоналу інтерактивного групового голосування за методом Кондорсе та Борда.....	39
ВИСНОВКИ	43

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ	45
ДОДАТОК А	48
ДОДАТОК Б.....	53
ДОДАТОК В.....	57



ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

1. AJAX – Asynchronous JavaScript And XML (укр. Асинхронний JavaScript та XML)
2. API – Application Programming Interface (укр. Прикладний програмний інтерфейс)
3. CSS – Cascading Style Sheets (укр. Каскадні таблиці стилів)
4. CCVS – Central Count Voting Systems (укр.)
5. CMS – Content Management System (укр. Система керування вмістом)
6. CLI – Command Line Interface (укр. Інтерфейс командного рядка)
7. DOM – Document Object Model (укр. Об'єктна модель документа)
8. DRE – Direct-recording Electronic (укр. Електронні машини прямого запису)
9. EBP – Electronic Ballot Printers (укр. Електронні принтери для голосування)
10. ES – ECMAScript
11. HTML – HyperText Markup Language (укр. Мова гіпертекстової розмітки)
12. HTTP – HyperText Transfer Protocol (укр. Протокол передачі гіпертекстових документів)
13. IDE – Integrated Development Environment (укр. Інтегроване середовище розробки)
14. OMR – Optical mark recognition (укр. Оптичне розпізнавання знаків)
15. PCVS – Precinct count voting systems(укр. Системи голосування на дільницях)
16. PNDRE – Public network DRE voting systems(укр. Голосування з прямим записом у публічній мережі)
17. PVS – Paper-based Voting Systems (укр. Паперові системи голосування)

18. SERVE – Secure Electronic Registration and Voting Experiment (укр. Безпечна електронна реєстрація та голосування)
19. SPA – Single-page application (укр. Односторінковий застосунок)



ВСТУП

Актуальність та постановка проблеми.

Однією з фундаментальних проблем прийняття групового рішення є неможливість прийти до згоди, коли більшість членів груп мають зовсім різні погляди та не можуть прийняти спільного рішення. Очевидно, якщо групу складається з великої кількості людей, які мають свої упередження та погляди, їх думки суттєво розділяються і унеможливають прийняття колективного якісного обґрунтованого рішення. За використання різних методів голосування результати можуть кардинально відрізнятись один від одного. Зокрема, якщо говорити про сферу освіти в Україні, розвиток технологій, який стрімко зростає з кожним роком та потребує все більш швидкої реакції сторони законодавства та ініціативи самих середніх та вищих навчальних закладів для впровадження інтерактивної освіти для школярів та студентів.

Актуальність роботи полягає в тому, що студенти потребують конкретної та необхідної інформації, яка надається лекторами під час навчального процесу, саме тому дана робота присвячена розробці програмного забезпечення, яке дає можливість ефективної комунікації між лекторами та студентами під час заняття стосовно наповнення розглянутого матеріалу, його якості та необхідності у подальшому розвитку студентів, у зв'язку з тим, що технології, які застосовуються у щоденному житті постійно змінюються та потребують фахівців, які повинні знати не лише теоретичну та історичну базу розглянутої області у рамках певного предмету, а й новітніх підходів та можливості розвиватися з практичної сторони протягом навчання у ВНЗ.

Мета роботи полягає у розробці інтернет ресурсу для запровадження системи організації інтерактивного голосування серед студентів задля покращення якості викладання матеріалу викладачами та миттєвої взаємодії між студентом та лектором.

Згідно поставленої мети, необхідно вирішити наступні **дослідницькі завдання:**

- Розглянути системи голосування, види, методи та використання кожного з них, переваги та недоліки групового голосування проти досягнення консенсусу, електронні системи голосування та їх різновидності, якими користуються сьогодні.
- Розглянути засоби реалізації застосунку, визначити мову програмування, фреймворк та бібліотеки, що використовуватимуться в проєкті, архітектуру та інтегроване середовище розробки, яке задовольнятиме потребам розробника.
- Реалізувати програмне забезпечення, веб-ресурс, для проведення інтерактивного голосування для студентів для покращення викладання та наповнення теоретичного матеріалу викладачами, реалізувати інтерактивне проведення групового голосування, що базується на методі Кондорсе та Борда.

Об'єкт дослідження – системи інтерактивного голосування.

Предмет дослідження – системи інтерактивного голосування для миттєвої комунікації між студентами та викладачами для покращення якості викладання матеріалу лектором.

Методологічна база дослідження.

У дослідженні було використано загальнонаукові методи, з яких: методи аналізу та синтезу існуючої методологічної бази про методи електронного та ручного групового голосування.

методи індукції та дедукції – у даній роботі описано системи електронного голосування, які з кожним роком стають все досконалішими та застосовуються у найрізноманітніших сферах життя, зокрема і в освіті, а також методи, на основі яких розробляється програмне забезпечення для інтерактивного обрахунку групових голосувань; використано парадокси Кондорсе та Борда для розробки інтерактивного голосування для практичних занять застосування.

метод моделювання ідеальних моделей;

емпіричний експериментальний метод та його евристична функція – на основі створеного веб-сайту реалізовано інтерактивне голосування для студентів стосовно якості та наповнення лекційного матеріалу під час лекційних занять та

можливість проводити інтерактивне голосування на практичних заняттях на основі методів Кондорсе та Борда.

Структура та обсяг роботи.

Робота складається зі анотації, змісту, переліку умовних позначень, символів, одиниць, скорочень та термінів, вступу, трьох розділів, чотирнадцяти підрозділів, висновків, списку використаних посилань та трьох додатків.

У першому розділі «Аналіз та методи систем голосування в групах» зосереджено увагу на теоретичній базі групового голосування, яка включає в себе теорію голосування та різноманітність існуючих систем голосування, найвідоміші методи голосування, до яких належать голосування простої більшості, метод Кондорсе, парадокс Борда; розглянуто поняття схвального голосування, методи пошуку консенсусу, а також наведено переваги та недоліки голосування проти досягнення консенсусу в групі, які перешкоди можуть виникати за використання кожного з них, а також наведено типологію електронних систем голосування, які використовуються сьогодні.

У другому розділі «Вибір засобів реалізації програмного забезпечення для електронного голосування» розглянуто вибір мови програмування для створення проекту, яке буде розроблено протягом написання бакалаврської кваліфікаційної роботи; порівняння інтегрованих середовищ розробки та обрано за допомогою якою буде відбуватись розробка застосунку, розглянуто фреймворк Angular2+ та його переваги для створення програмного забезпечення; проведено огляд архітектури проекту та перший запуск та ініціалізація проекту, що для цього потрібно та основні команди для розробника, який налаштовує середовище для початку роботи з Angular2-проектом; зокрема, розглянуто мову програмування TypeScript, на якій працює і побудований даний фреймворк, яка під час збирання проекту компілюється в JavaScript.

У третьому розділі «Реалізація інтерактивного голосування в групах для ефективної комунікації на лекційних заняттях у вищих навчальних закладах» розглянуто етапи ініціалізації проекту та необхідні для нього компоненти, до яких входить середовище виконання Node.js та додаток і сховище для розробки

NPM; наведено приклади UML-діаграм розробленого програмного продукту та описано реалізацію функціоналу для ефективної комунікації на лекційних заняттях у вигляді веб-ресурсу, а також додатковий функціонал для інтерактивного групового голосування за методом Кондорсе та Борда.

У висновках підбито підсумки виконання завдань, які було передбачено метою роботи та реалізовано веб-ресурси для інтерактивного голосування в групах під час лекційних та практичних занять студентів.

Список використаних джерел складається з 19 інтернет ресурсів та 5 наукових статей та книжок .



РОЗДІЛ 1. АНАЛІЗ ТА МЕТОДИ СИСТЕМ ГОЛОСУВАННЯ В ГРУПАХ

1.1. Теорія голосування. Різновиди систем голосування

У сучасному суспільстві цифрові послуги стали важливою частиною життя. Електронне голосування застосовується у багатьох країнах світу. Щороку, за останні два десятиліття, кількість теоретичних та практичних рішень та наукових робіт у цій галузі стрімко збільшується. Існує наукова стаття, яка вважається першопрохідцем у цій галузі, у ній розглядаються схеми для електронного голосування Девіда Чаума. Основна увага в області електронного голосування найчастіше приділяється технічним вимірам: криптографічним алгоритмам, протоколам та схемам електронного голосування, надійності обладнання, впровадженню програмного забезпечення, безпеці тощо. Останнім часом фокус перейшов до організаційних, соціальних та політичних аспектів електронного голосування. Незважаючи на велику кількість досліджень у галузі електронного голосування, станом на сьогодні, воно досі не має великих успіхів у реальному світі. Електронні машини прямого запису (DRE) для електронного голосування неодноразово критикували. Міністерство оборони США запропонувало дистанційну систему голосування на виборах в Інтернеті: Безпечна електронна реєстрація та голосування (SERVE), але у їхньому звіті є настійні рекомендації проти використання даної системи.

Існує ціла низка систем голосування, серед яких найпопулярнішими є [1]:

1. Паперові системи голосування (PVS)

Дана система дозволяє записувати або реєструвати, підраховувати голоси, подані на паперових картках або аркушах. Деякі з паперових систем дають можливість здійснювати голосування за допомогою електронних пристроїв. Однак, виборці не мають змоги зареєструватись самостійно або внести свої дані в загальну таблицю голосів за допомогою власних електронних пристроїв.

2. Електронні системи голосування з прямим записом даних (DRE)

Такі системи дають можливість у режимі реального часу голосувати за допомогою різних девайсів, які за допомогою вбудованих оптичних елементів, що мають бути активовані під час проходження голосування на персональному комп'ютері або іншому девайсі. Дані електронні системи записують голос або зображення у пам'яті комп'ютера, а також обробка даних стає можливою лише за використання спеціалізованих комп'ютерних програм.

3. Системи голосування з прямим записом у публічній мережі (PNDRE)

Такі системи дають можливість також у режимі реального часу передавати дані через загальнодоступну мережу, за допомогою чого проголосувати можна навіть з дому, а голоси будуть передаватись як окремі виборчі бюлетені від кожного особи персонально або їх можуть об'єднувати у великі партії після офіційного завершення часу голосування.

4. Системи голосування на дільницях (PCVS)

Подібні системи забезпечують можливість за допомогою спеціальних зчитувачів обробляти дані голосування одразу на дільниці та надсилати до центрального пункту отримання всіх результатів.

5. Системи центрального голосування (CCVS)

Бюлетені з декількох дільниць направляють в одне місце, де вони підраховуються, іноді, за використання даної системи, можуть бути оприлюднені друковані звіти після проведення повного підрахунку голосів.

В суспільстві голосування є формальним способом висловлення думок. Відомим прикладом є обрання президента. У цьому прикладі виборці – це люди, які висловлюють свою думку за допомогою голосування. Під час голосування учасник вибирає одного з кандидатів або вказує певний порядок, який відображає рейтинг його переваг. Метод голосування – це механізм інтегрування всіх голосів в одне остаточне рішення. Переможцем стає той кандидат, який обраний за результатами методу голосування.

Теорія голосування – це математична дисципліна, яка зосереджена на розробці «чесних» процедур голосування. Визначено ряд критеріїв справедливості [2]:

1. Критерій більшості: переможцем може стати будь-який кандидат, який отримав більшість голосів за перше місце.
2. Критерій Кондорсе: переможцем стає той, хто виграє у голосуванні один на один з усіма іншими кандидатами.
3. Критерій монотонності: якщо відбудуться вибори та буде оголошено переможця, кандидат-переможець залишатиметься переможцем у будь-якій ситуації, незважаючи на перерахунок голосів на користь інших кандидатів.
4. Критерій незалежності нерелевантних альтернатив: якщо відбудуться вибори та буде оголошено переможця, кандидат-переможець залишатиметься переможцем у будь-якій ситуації, незважаючи на перерахунок голосів на користь інших кандидатів. Аналогічно до критерію монотонності.

1.2. Методи систем голосування

Існує безліч методів голосування, які використовуються на виборах, законодавчих процесах та груповому прийнятті рішень, до яких входять метод голосування простої більшості, схвальне голосування, метод Борда, метод середнього балу та інші.

Наявність великої кількості систем голосування свідчить про те, що люди в різні часи мали дещо різні уявлення про те, як слід робити колективний вибір. Або вони, можливо, хотіли зробити акцент на дещо різних аспектах процесу вибору. Бінарні системи, як правило, підкреслюють, що кінцеві переможці будуть обрані за Кондорсе. Якщо припустити, що виборці голосують за ту партію, яка лідирує у рейтингу, може статися так, що переможець за Кондорсе буде обраний вже на початкових етапах процесу. Також позиційні системи голосування, наприклад, плюралістичне голосування і підрахунок методом Борда можуть не визначити переможця за Кондорсе. Сильним переможцем Кондорсе є альтернативний варіант, який займає перше місце серед більш ніж половини електорату. Велика кількість систем задовольняє даний критерій. Єдиним винятком є підрахунок голосів методом Борда та схвального голосування.

У 1995 році Дональд Саарі представив власний геометричний підхід до систем голосування. Його репрезентативні трикутники, або їх ще називають трикутниками Саарі, корисні для ілюстрування впливу різних компонентів на голосування (рис. 1.1) [3].



Рисунок 1.1 – Приклад репрезентативного трикутника Саарі [3]

У наведеному прикладі репрезентативного трикутника (див. рис. 1.1) все вказує на лідера А: це переможець у множинності сильний переможець за Кондорсе. Проте це не переможець за методом Борда. Переваги над трьома альтернативами можна перевести в односторонній трикутник з вершинами, що стоять за альтернативами. Намалювавши всі внутрішні лінії всередині трикутника, вийде 6 маленьких трикутників. Кожен з них представляє рейтинг переваг, так що відстань від вершин визначає рейтинг. Отже, область, позначена 7, представляє рейтинг АВС, оскільки А найближча до вершини А і ближча до В, ніж С. Аналогічним чином, вершина трикутника, позначена 4, найближча до вершини В, а С є наступною.

Переможців за Борда та Кондорсе можна визначити з репрезентативного трикутника наступним чином. Сума двох записів у трикутниках, найближчих до кожної вершини, дає множину голосів кандидата, представленого вершиною. Таким чином, наприклад, $7 + 0$ – це множина голосів А. Показник А за методом Борда, в свою чергу, можна обчислити шляхом підсумовування записів у лівій частині відрізка, що з'єднує С та середню точку лінії АВ, а також записи на нижній стороні відрізка, що з'єднує В і середню точку лінії АС. Тобто $7 + 7 = 14$. Подібним чином, оцінка за методом Борда для В становить $11 + 4 = 15$, а для С –

$4 + 4 = 8$. Про те, що A – переможець за Кондорсе, можна зробити висновок з того, що обидва його підсумки перевищують 5,5, кількість виборців розділена на два. Про те, що C є переможеним за Кондорсе, можна зробити висновок і з його підсумків: вони обидва менше, ніж більшість виборців.

1.2.1. Метод голосування відносної більшості

Голосування відносної більшості або метод множинності широко використовується серед різноманітних методів голосування, оскільки він є простим та інтуїтивним. Кожна людина в системі вибору має один голос. Кожен з виборців схвалить найбільш бажану альтернативу. Перемагає альтернатива, яка набрала найбільше голосів. Перш за все, якщо існує після підрахунку результатів залишається більше, ніж одна альтернатива, буде проведена додаткова процедура для визначення переможця. Однією з проблем даного методу є поняття «розколу більшості». Двоє популярних кандидатів можуть розділити підтримку виборців, таким чином дозволяючи третьому кандидату, який має лише підтримку меншості, стати переможцем. Багато методів голосування є варіаціями методу множинності. Деякі з них зобов'язують переможця перейти певний поріг, щоб стати першим серед інших. Поріг може коливатися від половини голосів до загальної підтримки виборців (одностайної підтримки). Виборці повинні голосувати повторно, поки одна з альтернатив не перевищить поріг. Ще одним із різновидів методу простої більшості є «другий тур» голосування, тобто виключення альтернатив із наступного туру до того, як буде остаточний переможець. Однією з версій повторного голосування є надання дозволу двом альтернативам, які мають найбільшу підтримку, пройти до другого туру голосування. Інший варіант другого туру виключає альтернативу з найменшою кількістю голосів у кожному турі. Процес повторюється, поки не буде остаточно виграшної альтернативи (1.1) [2].

$$PI(A, >) = \{a \in A: n_1(a) \geq n_1(b) \text{ для всіх } b \in A \setminus a\} \quad (1.1)$$

1.2.2. Метод Кондорсе

Метод Кондорсе – це система голосування, яка служить для обрання переможця серед групи кандидатів. Виборці виставляють свої бали за певного кандидата по спадаючій, тобто найбільший бал надається тому кандидату, до якого найбільше схиляється виборець, а далі все відбувається за спаданням, відповідно найменш привабливому для виборця кандидату віддається найнижчий бал [3].

Основні характеристики використання методу Кондорсе:

- Розподіл балів між кандидатами відбувається від найбільшого балу до найменшого.
- Необхідно порівнювати кандидатів, той хто має найбільшу перевагу для виборця, за того він і голосує, відповідно і отримує голос.
- Перемагає той кандидат, сума голосів у перемогах якого буде найбільшою серед інших кандидатів.

Зокрема, після появи даного методу було розроблено низку інших для підтвердження переможця за методом Кондорсе, до них входять:

1. Процедура Блека – кожен виборець представляє власний рейтинг кандидатів, якщо його кандидат є переможцем за методом Кондорсе, тоді він справді переможець, інакше необхідно використовувати метод Борда.
2. Метод Нансона – кожен виборець представляє власний рейтинг кандидатів, для кожного з них обчислюється оцінка за методом Борда і ті кандидати, які отримали оцінку, що є нижче середньої, вибувають, а далі бали знову перераховуються серед тих, хто залишився і процес триває до того моменту, до поки не залишиться один кандидат.

1.2.3. Метод Борда (парадокс Борда)

Метод Борда враховує весь рейтинг індивідуальних уподобань при виборі альтернатив, де для кожної проводиться підрахунок на основі її рейтингу в уподобаннях окремої людини. Для n альтернатив найчастіше використовується спосіб присвоєння підрахунків альтернативам – це $n-1$ бал за кожен бюлетень, за

яким він займає перше місце, $n-2$ за друге і так продовжується до останнього кандидата у рейтингу. Виграє альтернатива з найбільшим загальним підрахунком. Варіації використання методу Борда можуть мати в собі можливість призначення різної ваги різним рангам. Незважаючи на те, що даний метод є приводом для дискусій вченими протягом століть, не існує єдиної думки щодо того, який переможець буде «кращим» [3].

Підрахунок за методом Борда полягає у тому, що кожен виборець заявляє про свої переваги відповідно до рейтингу кандидатів p зверху вниз. При виборі з n кандидатів кожен голосуючий ранжує всіх кандидатів виключно по спаданню переваг, де за перше місце кандидату присуджують n балів, за друге $n-1$ балів і так до 0, що вже було розглянуто вище. За останнє місце присуджують 1 бал, всі набрані кандидатами бали сумуються. Відповідно, переможцем виборів рахується кандидат, який набере найбільшу кількість балів.

Отже, даний парадокс, метод або його ще називають правилом Борда, опирається на бали і визначається за формулою (1.2):

$$Bo(\alpha) = \sum_{i=1}^m n_i(\alpha)(m - i), \quad \alpha \in A \quad (1.2)$$

Перемагає той кандидат, який набрав найбільшу кількість балів. А бали можна підрахувати, використовуючи зведену таблицю голосів (1.3).

$$Bo(\alpha) = \sum_{b \in A \setminus \{\alpha\}} h(\alpha, b), \quad \alpha \in A \quad (1.3)$$

Одним з недоліків даного методу вважають, що переможець обраний за методом Борда, не завжди буде займати ту ж позицію у рейтингу при використанні методу Кондорсе. У парадоксі Борда присутні наступні особливості [4]:

1. Ступінь висловлення суб'єктивних думок і використання таких даних для прийняття важливих рішень впливає на об'єктивність загального результату оцінювання. Хоча для множинного голосування потрібна лише мінімальна кількість інформації про позиції виборців, існують методи, зокрема метод Борда, який допомагаю представити набагато яскравішу форму висловлення думок. Таким чином, це ставить під питання «правильність» форми висловлювання думок виборцями.

2. Існує декілька концепцій перемог, попарно та позиційно. «Перемога» може означати різні речі для різних людей. Ідея, що лежить в основі плюралістичного голосування, згідно з якою, найбільш ймовірним кандидатом на перемогу є той, хто розглядається зі сторони позиційності.

3. У концепціях позиційності та попарності використовуються несумісні між собою погляди на перемогу, що впливає на перевірку об'єктивності та реальності прийнятого рішення.

4. За умови обрання абсолютною більшістю кандидата з найвищим рейтингом, то переможці як попарні, так і обрані множинною перевагою збігаються, але він посідає перше місце лише через визнання його переможцем абсолютною більшістю.

1.2.4. Схвальне голосування

Схвальне голосування було створено для подолання проблеми, де лише найвища альтернатива у списку переваг людини розглядається методом множинності. У процедурі голосування за схвалення кожен виборець у системі може віддати лише один голос серед всіх альтернатив, які він або вона схвалює. Переможцем оголошується альтернатива, яка набрала найбільше голосів. Схвальне голосування не відповідає критерію перемоги Кондорсе та умові Парето. Переважно, даний метод використовується лише за умови, що переможець повинен бути один, проте його також можна застосувати за умови ймовірності вибору двох переможців. Схвальне голосування не має різноманіття у голосах або балах, які віддають за кандидата або альтернативу, у ньому існує лише два розвитку подій, серед яких або надати перевагу, або ні [2].

Основною відмінністю даного методу, є те, що в нього є обов'язковим використання бюлетенів, проте в них можна обирати не одного кандидата, як зазвичай, а декількох, тому для цього вносяться незначні зміни для стандартів друку бюлетенів, якщо мова іде про паперове голосування. У США проводили дослідження стосовно застосування схвального голосування, було проведено експеримент на прикладі екзит-полу на офіційних виборчих дільницях у штаті

Нью-Йорк та безпосередньо в одному з округів на Манхетені. У даному експерименті було проведено порівняння методу відносної більшості, або множинного голосування, де виборці мають право віддати свій голос лише за одного кандидата, з методом схвального голосування. За результатами опитувань, переможець не змінився, проте кількість голосів за другого кандидата зросла на 15% з підходом схвального голосування на противагу множинному, а також інші партії та кандидати отримали більший відсоток голосів [5].

1.2.5. Методи пошуку консенсусу

Існує безліч методів голосування для досягнення очевидно ідентичного результату – виокремлення співвідношення колективних переваг, – враховуючи той факт, що методи нееквівалентні. Причини їх появи досить відрізняються один від одного. Однак, можна стверджувати, що в основі методів є спільні риси, а саме: ідея консенсусу. Більше того, кожен метод базується на ідеї мінімізації відстані, вимірюваної певним чином, між наданням переваги та переходом до консенсусу. Якщо ця ідея спільної домовленості буде прийнята, тоді можна розглядати безліч методів, зважаючи на різницю думок щодо поняття консенсусу, а також на реалізацію, за якої це можна втілити у життя. Існує метод, який явно базується на вищезазначеній ідеї мінімізації відстані: правило Кемені. Розглядаючи кандидата у якого більше прихильності, він визначає рейтинг переваг для всіх альтернатив, які є найближчими до спостережуваного, у тому сенсі, що для отримання цього рейтингу потрібна мінімальна кількість змін. Таким чином, консенсус, від якого вимірюється відстань до спостережуваного кандидата в системі Кемені, – це одностайність щодо всіх позицій у рейтингу альтернатив, тобто виборці домовляються про те, яка альтернатива розміщується першою, яка друга і так до останньої [3].

1.3. Переваги та недоліки голосування проти досягнення консенсусу

Починаючи працювати у новому колективі, неважливо це навчання або робота, важливо продумати, як колектив буде приймати рішення – від великих

до звичайних тривіальних рішень. Прийняття будь-яких рішень у групах може відбуватись за рахунок голосування або за досягнення консенсусу стосовно деякого рішення (табл. 1.1). Зокрема, можна використовувати гібридний підхід – запропонувати групі проголосувати за менш вагомим рішенням, будуючи консенсус щодо більш важливих. Який би підхід – або комбінацію підходів – не було б обрано, важливо, щоб даний процес проходив прозоро та кожен учасник процесу чітко усвідомлював свої та дії інших.

Таблиця 1.1 – Порівняльна характеристика використання голосування або досягнення консенсусу [6]

Метод	Голосування	Консенсус
Опис	Більшість приймає рішення (з групою спочатку визначають, що та хто буде «більшість»).	<ul style="list-style-type: none"> ● Кожен озвучує свою позицію. ● Усі члени команди погоджуються підтримати остаточне рішення, навіть коли це не вибір більшості.
Переваги	<ul style="list-style-type: none"> ● Знайоме, звичайне, ефективне 	<ul style="list-style-type: none"> ● Забезпечує підтримку всіх членів, тим самим
	<ul style="list-style-type: none"> ● Вибір більшості вважають «справедливим» ● Корисно, коли обмеження у часі ускладнюють досягнення консенсусу 	<ul style="list-style-type: none"> ● збільшуючи ймовірність успіху. ● Учасники прагнуть прийняти найкраще рішення для групи, а не змагатись за особисті уподобання. ● Всі перспективи враховуються.
Недоліки	<ul style="list-style-type: none"> ● Не дає можливості обговорювати думки меншості ● Може не заохочувати взаємодію зі всією групою ● Створює переможців та переможених 	<ul style="list-style-type: none"> ● Трудомісткий ● Чим більша група, тим складніше реалізувати ● Можуть виникати спірні питання ● Іноді нудно

Продовження таблиці 1.1

Варіації	<ul style="list-style-type: none"> ● Плюралізм (вибір більшості): велика кількість учасників підтримує одне рішення. ● Ранжування: кожен учасник голосує за кожен варіант, а тоді виграє той, в якого найвищий середній бал. 	<ul style="list-style-type: none"> ● Від одного до п'яти: члени команди використовують специфічні жести: щоб заблокувати консенсус (кулак), запропонувати зміни (один палець), обговорити незначні проблеми (два пальці), пройти без обговорення (три пальці), підтвердити рішення (чотири пальці), або добровольцем взяти на себе ініціативу у виконанні нового рішення (п'ять пальців). ● Червоні/жовті/зелені кольорові картки: червона картка вказує на протидію рішенню, жовта картка – застереження, а зелена – згоду
----------	--	---

1.4. Типологія електронних систем голосування

Усі типології електронного голосування мають різні сильні та слабкі сторони як у порівнянні між собою, так і у порівнянні з традиційним голосуванням. Не існує ідеальної електронної системи голосування, і наявні системи продовжують розвиватися. Тому важливо правильно обирати систему для відповідного контексту, ретельно зважуючи переваги та недоліки всіх варіантів. Загалом, будь-яку систему електронного голосування можна віднести до одного з наступних типів [7]:

1. Електронні машини для прямого запису (DRE). Вони використовують паперові бланки для обробки результатів голосування.
2. Системи оптичного розпізнавання знаків (OMR) функціонують за допомогою сканерів, які можуть розпізнати вибір виборців на спеціальних бюлетенях, які може прочитати комп'ютер. Системи OMR можуть бути як

центральною системою підрахунку голосів, де бюлетені скануються та підраховуються в спеціальних центрах підрахунку голосів, так і системами оптичного сканування дільничних підрахунків, де сканування та підрахунок відбувається на виборчій дільниці безпосередньо, коли виборці залишають свої бюлетені в машинах для голосування.

3. Електронні принтери для голосування (EBPs) – це пристрої, подібні до машин DRE, що виробляють папір для машинного читання або токен, в який перетворюються дані голосування окремого виборця. Цей токен відправляється в окремий сканер виборчих бюлетенів, який робить автоматичний підрахунок голосів.

4. Інтернет-системи голосування, де голоси передаються через Інтернет на центральний сервер підрахунку голосів. Голоси можна надсилати як із домашніх комп'ютерів загального користування, так і з виборчих дільниць, тобто з будь-якого девайсу, що має доступ до мережі Інтернет та яким може користуватися виборець особисто.

Електронне голосування може проводитися як в контрольованому, так і в неконтрольованому середовищі. Електронне голосування в контрольованому середовищі відбувається, коли голосування проводиться на виборчих дільницях чи інших місцях під наглядом персоналу, призначеного органом управління виборами. Таким чином, виборча адміністрація може значною мірою контролювати процес голосування, а також процедури та умови, за яких виборці віддають свої голоси. Електронне голосування в контрольованому середовищі може розглядатися як електронний еквівалент традиційного голосування на папері на виборчих дільницях, в посольствах тощо [8].

Електронне голосування в неконтрольованому середовищі відбувається без будь-якого нагляду та за допомогою пристроїв для голосування, які не можуть контролюватися адміністрацією. Це може бути вдома, на персональному комп'ютері виборця або потенційно в будь-якому місці на мобільних або громадських пристроях. Голосування в неконтрольованому середовищі може бути не надійним, якщо розглядати даний процес зі сторони таємниці

голосування, сімейного голосування, залякування, купівлі голосів, наслідків цифрового впливу, а також, оскільки, технічна надійність пристрою, за допомогою якого надсилається голос, потребує бути надійним. Сучасні форми голосування в Інтернеті ще не змогли остаточно вирішити такі проблеми. Електронне голосування в неконтрольованому середовищі може розглядатися як електронний еквівалент голосування поштою або заочного голосування.



РОЗДІЛ 2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЕЛЕКТРОННОГО ГОЛОСУВАННЯ

2.1. Вибір мови програмування

На сьогоднішній день існує велика кількість мов програмування, у даному проєкті було використано мову програмування JavaScript.

Тривалий час сайти працювали на основі PHP CMS, таких як WordPress, де серверний код обробляв основну частину логіки.

Більшість веб-застосунків створені за допомогою JavaScript. За допомогою цієї мови програмування користувач може взаємодіяти з інтерфейсом сторінки, до прикладу соціальна мережа Facebook дає можливість заповнити власний профіль, редагувати та оновлювати дані.

На основі JavaScript було побудовано такі фреймворки як Angular та Vue, а також бібліотеку з відкритим кодом React.

Розглянемо односторінкові застосунки (SPA) – це сайти, які взаємодіють з користувачами, динамічно оновлюючи сторінку в браузері, а не завантажуючи нові з сервера, змушуючи їх поводитися більше як настільні програми [9].

Станом на сьогодні, JavaScript може використовуватись не тільки для веб-розробки, а також дає можливість реалізовувати наступний функціонал:

- Серверний код. JavaScript-розробники можуть використовувати Node.js для написання клієнтської частини коду, так і серверну.
- Інтеграція API. Розробники можуть використовувати JavaScript для отримання даних з інших джерел та відображення їх на власному сайті. Однією з концепцій, яка як ніколи популяризується у веб-розробці, є модульність – використання різних інструментів для виконання конкретних завдань.

Переваги Java Script [10]:

1. JavaScript дає можливість створювати адаптивні сайти. Дедалі більше розробникам потрібно адаптувати свій дизайн у декількох браузерах та пристроях. Поєднуючи HTML5, CSS3 та JavaScript, вони можуть це зробити в одній кодовій базі.

2. Для початківців ця мова є легкою для вивчення і дозволяє активно розвиватись. Його синтаксис простий і гнучкий для новачків. Це також спрощує розробку складних додатків, дозволяючи спростити склад програми. Багато фреймворків та пакетів також полегшують роботу розробникам.

3. Швидкість. Клієнтський JavaScript дуже швидкий, оскільки його можна негайно запустити в клієнтському браузері.

4. Сумісність. JavaScript чудово поєднується з іншими мовами і може використовуватися у великій кількості різноманітних додатків.

5. Навантаження сервера. Виконуючись на стороні клієнта, зменшується кількість запитів на сервер сайту.

2.2. Інтегровані середовища розробки

Сьогодні є безліч популярних середовищ розробки (IDE), серед яких можна виділити декілька самих популярних.

1. Visual Studio Code – це безкоштовний редактор коду, який допомагає розробнику писати, налагоджувати та виправляти написаний код, за допомогою методу intellisense.

Visual Studio Code має низку корисних функцій [11]:

1. Підтримка декількох мов програмування
2. Intelli-Sense дає можливість виявляти незавершені фрагменти коду.

Крім того, синтаксис змінних та їх оголошення робиться автоматично. Наприклад, якщо в програмі використовується певна змінна, і користувач не оголосив її, Intelli-sense зробить це замість нього.

3. Підтримка терміналів.

4. Мультипроекти. Можна одночасно відкривати декілька проектів. Ці проекти можуть бути пов'язані між собою або ніяк не пов'язані.

5. Підтримка Git. Файли можна зручно відправляти та отримувати з будь-якого хмарного репозиторію, використовуючи лише власний акаунт, а саме логін та пароль від нього. Витягування ресурсів також означає клонування коду, який доступний в Інтернеті. Цей код можна згодом змінити та зберегти.

6. Кросплатформеність. Раніше інтегровані середовища були написані для використання лише на певній операційній системі, проте перевагами сучасних середовищ є те, що їх можна встановити та використовувати з однаковим функціоналом як на Windows, так на Linux і Mac. Але Visual Studio Code є кросплатформним середовищем розробки.

7. Легкий для розуміння. Зручність використання є однією з найбільших переваг для розробника, файли впорядковуються ієрархічно і мають звичайні програмні засоби, такі як панель інструментів, рядок стану та бічна панель. Зокрема, середовище має плаваюче вікно провідника, яке можна закріпити в будь-якому зручному для використання місці, яке складається зі структури каталогів файлів. Ці файли (файли коду, папки зображень тощо) можна відкрити або перейменувати одразу, а зміни автоматично відобразяться у сховищі.

2. Sublime – швидкий кросплатформний, частково безкоштовний текстовий редактор. Підтримує плагіни, розроблені за допомогою мови програмування Python. Sublime Text має платний та безкоштовний функціональний набір, але деякі його плагіни розповсюджуються з вільною ліцензією, розробляються і підтримуються спільнотою розробників. Редактор містить різні візуальні теми, з можливістю завантаження додаткових з мережі Інтернет. Користувачі бачать весь свій код в правій частині екрану у вигляді міні-карти, при натисканні на яку можна взаємодіяти з навігацією. Є кілька режимів екрану. Один з них включає від 1 до 4 панелей, за допомогою яких можна показувати до чотирьох файлів одночасно. Повноцінний (free modes) режим показує тільки один файл без жодних додаткових панелей [12].

3. WebStorm – це середовище для розробки із використанням мови JavaScript. Хоча це легкий інструмент, він містить комплексні функції для складної розробки на стороні сервера та клієнта за допомогою Node.js. Додаток, створений компанією JetBrains та використовує потужну екосистему JavaScript і надає розробникам корисні функції, такі як мовна таблиця стилів, рефакторинг та навігація, виявлення помилок у реальному часі тощо. Система дозволяє

користувачам легко проводити тестування Node.js на стороні клієнта на наявність помилок, а потім виправити їх. Він також використовує інтерактивну консоль. Існують різні інструменти, які легко інтегруються в систему. Функціональні можливості модульного тестування дозволяють користувачам налагоджувати тести, які виконуються на Protractor, Karma, Jest та Mocha. Статус тестів негайно відображається як у редакторі, так і у перегляді дерева [13].

2.3. Переваги використання Angular

В даний час при розробці веб-застосунків майже неможливо обійтись без фреймворків, таких як Angular, Vue чи бібліотеки React. Хоч вони і виконують, на перший погляд, одні й ті самі задачі, вони зовсім різні.

Angular – це інтерфейсний фреймворк з відкритим кодом, розроблений компанією Google для створення динамічних сучасних веб-програм [22]. Angular допомагає створювати інтерактивні та динамічні односторінкові додатки (SPA) за допомогою своїх функцій, включаючи шаблонування, двосторонню прив'язку, модульність, обробку RESTful API, введення залежностей та обробку AJAX та HTTP запитів.

При роботі з Angular можна виділити наступні переваги [14]:

1. Підтримка Google. Однією з найбільших переваг Angular є те, що вона підтримується та постійно розширюється компанією Google. Це вказує на те, що Google буде і надалі підтримувати його і ще більше розширювати екосистему Angular. Додатки Google також використовують цей фреймворк, і їх команда дуже оптимістично оцінює стабільність Angular.

2. TypeScript. Angular додатки будуються за використання мови програмування TypeScript. TypeScript – це строго типізована, об'єктно-орієнтована, компільована мова, що забезпечує вищий рівень безпеки, оскільки підтримує типи (примітиви, інтерфейси тощо). Це допомагає ловити та усувати помилки на початку написання коду. На відміну від CoffeeScript або Dart, TypeScript не є окремою мовою. За допомогою TypeScript є можливість легко взяти наявний код написаний на ES5 або ES6+, і він скомпілює його на основі

того, що буде вказано в конфігураційному файлі. Він повністю підтримує основні функції ES2015 та ES2016 / ES2017, такі як декоратори або `async / await`. TypeScript дає можливість відлагоджувати код у браузері або редакторі. Ця мова програмування забезпечує використання навігації, рефакторингу та автозаповнення.

3. Декларативний інтерфейс. TypeScript, порівняно з JavaScript, є менш заплутаною мовою. Це допомагає тим, що не потрібно визначати те, що завантажується першим.

4. Легке тестування.

5. Модульна структура. Модулі полегшують організацію функціональних можливостей програми, розділяючи її на функції та компоненти багаторазового використання. Модулі також дозволяють ліниве завантаження, що дає можливість завантажувати частину сайту пізніше, тоді коли ця частина буде необхідна.

Angular дає можливість розподілити роботу між різними членами команди. Це можливо реалізувати через те, що сайт написаний за допомогою Angular складається з різних компонентів.

2.4. Архітектура Angular-проекту

Архітектуру програмного забезпечення можна визначити як «розбиття системи на частини». Зазвичай, архітектурні рішення, які приймаються при розробці системи, важко змінити, оскільки вона постійно розширюється. Саме тому дуже важливо зважено приймати рішення стосовно вибору найбільш відповідної архітектури, особливо, якщо програмне забезпечення, яке створюється, має працювати протягом багатьох років.

Angular-проект поділяється на шість основних блоків [15].

1. Модулі. Кожен застосунок, написаний на Angular, має кореневий модуль, який має назву – `AppModule`. Він складається з підключення та оголошення модулів, компонентів та бібліотек, які будуть використані у проекті, а також дає змогу налаштувати навігацію між різними сторінками веб-

застосунку. Якщо розробник хоче використовувати інший спеціальний модуль, тоді йому потрібно його зареєструвати у файлі `app.module.ts`. Організація коду з використанням окремих функціональних модулів допомагає керувати розробкою складних додатків, а також дає можливість повторного використання.

2. Компоненти. Кожен Angular-проект має принаймні один компонент, який створюється при ініціалізації проекту та має назву `app.component`, крім того, при створенні компонентів ініціалізується одразу чотири файли з такими розширеннями як `.html`, `.css` (за використання препроцесорів розширення буде відрізнятися), `.spec.ts` та `.ts`.

3. Шаблони. Angular-шаблони поєднують HTML-розмітку з Angular-розміткою, яка може змінювати елементи HTML до моменту їх відображення на сторінці.

4. Метадані. Метадані передають інформацію Angular-проекту стосовно поведінки та обробки класу компоненту, в якому він оголошений. Вони використовуються для розширення можливостей класу, щоб розробник міг налаштувати його поведінку. Декоратори – це основна концепція розробки при використанні Angular (версії 2 і вище). Користувач може використовувати метадані в класі, щоб повідомити програму, що `AppComponent` є компонентом. Метадані можна приєднати до TypeScript за допомогою декоратора [16].

5. Прив'язка даних. Прив'язка даних відіграє важливу роль у зв'язку між шаблоном та його компонентом. Прив'язка даних також важлива для зв'язку між батьківськими та дочірніми компонентами. Angular дозволяє визначити зв'язок між компонентом і DOM, для швидкої передачі або отримання даних.

6. Директиви. Компонент Angular – це директива, яка використовує шаблон для розмітки сторінки. Компоненти є основними блоками Angular-застосунків, проте, директиви є фундаментальними частинами проекту, які допомагають покращувати та скорочувати кількість коду за допомогою можливості повторного використання функціоналу за допомогою підключення та оголошення директиви в необхідному компоненті.

Архітектура спирається на певні фундаментальні концепції. Основними елементами Angular-проекту є компоненти, які обробляються за допомогою NgModules. NgModules збирає відповідний код у функціональні набори, в свою чергу Angular-додаток складається з певного переліку NgModules. Проект завжди має кореневий модуль [16].

Компоненти, в свою чергу, містять розмітку, яка являє собою набір елементів, які за допомогою Angular можна відображати та модифікувати відповідно до функціональних та користувацьких потреб.

Компоненти використовують сервіси, які надають певні функціональні можливості, не пов'язані з відображенням. Сервіс може бути додано до компоненту як залежність, що розширює функціональні можливості компоненту, який стає модульним та дає можливість повторного використання.

Модулі, компоненти та сервіси – це класи, що використовують декоратори. Ці декоратори позначають свій тип і надають метадані, які вказують Angular на те, як ними користуватися.

Angular-модулі відрізняються від модулів JavaScript (ES2015) та доповнюють їх. NgModule оголошує контекст компіляції для набору компонентів, призначеного для домену програми, робочого процесу або тісно пов'язаного набору можливостей. NgModule може пов'язувати свої компоненти з відповідним кодом, таким як сервіси, для формування функціональних одиниць.

Як і модулі JavaScript, NgModules можуть імпортувати функціональні можливості з інших NgModules, а також дозволяють експортувати та використовувати власні функціональні можливості іншими NgModules. Наприклад, щоб використовувати сервіс навігації у Angular-додатку, необхідно імпортувати Router NgModule [17].

Організація коду в окремі функціональні модулі допомагає керувати розробкою складних додатків та розробкою для повторного використання. Крім того, ця техніка дозволяє скористатися перевагами ледачого завантаження, тобто завантаження модулів на вимогу, щоб мінімізувати кількість коду, який потрібно завантажувати під час запуску.

2.5. TypeScript в Angular-проекті

TypeScript є основною мовою для розробки додатків Angular. Станом на сьогодні, веб-браузери не можуть виконувати TypeScript без попередньої компіляції файлу в JavaScript за допомогою компілятора tsc, який потребує певної конфігурації [18].

Не лише TypeScript є мовою програмування, яка компілюється в JavaScript, наприклад, CoffeeScript, Ruby, PHP та багато інших. Проте, на практиці більшість з них не мають нічого спільного з JavaScript, окрім компілятора. Для того, щоб перенести код з JavaScript на TypeScript необхідно вибрати модуль, перейменуйте файли .js у .ts, а потім оголошувати типи даних для різних змінних.

Сьогодні існує безліч версій та підходів до використання JavaScript, до яких належать: ES5, ES6 (Babel), TypeScript, Dart, PureScript, Elm тощо.

ES5 має одну суттєву перевагу перед TypeScript: для неї не потрібен інтерпретатор. Для початку роботи потрібно лише налаштувати програму перегляду файлів, переводити код, генерувати вихідні карти.

ES6 потребує інтерпретатора, тому налаштування збірки мало чим відрізнятиметься від TypeScript. Але це стандарт, який означає, що кожен окремий інструмент редактора та збірки або підтримує ES6, або зможе робити це в майбутньому. Проте, це примітивний приклад, який використовували раніше, оскільки більшість редакторів на даний момент мають підтримку TypeScript [18].

2.6. Структура робочої області та модулі JavaScript

Робоча область містить файли для одного або декількох проектів. Проект – це сукупність файлів, що містять автономну програму або бібліотеку, якою можна поділитися [15].

Angular CLI налічує низку команд, які можна прописувати в терміналі для автоматичного створення або налаштування необхідних елементів проекту, до них належать:

1. `ng new` – створює робочу область. Коли розробник запускає цю команду на виконання, CLI встановлює необхідні пакети прм та інші залежності

в новій робочій області з проектом на кореновому рівні під назвою `my-project`, якщо не було вказано іншої назви при написанні команди. Коренева папка робочої області містить різні файли підтримки та конфігурації, а також файл `README` зі згенерованим текстом, який можна змінити на власний розсуд. Команда `ng new my-app` створює папку робочої області з назвою `"my-app"` і генерує новий скелет програми в папці `src` на верхньому рівні робочої області. Щойно створена програма містить вихідні файли для кореневого модуля з кореневим компонентом та шаблоном. За замовчуванням, `ng new` створює проекту з на кореновому рівні робочої області з файлами для написання програмного тестування. Кореневий проект має те саме ім'я, що і робоча область, а вихідні файли знаходяться у папці `src` в робочій області.

Angular також підтримує можливість взаємодії декількох проектів одночасно. Цей тип середовища розробки підходить для досвідчених користувачів, які розробляють спільні бібліотеки, та для підприємств, які використовують стиль розробки `"monorepo"`, з єдиним сховищем та глобальною конфігурацією для всіх проектів Angular.

2. `ng generate` – використовується після ініціалізації проекту, дає можливість розширювати функціональні можливості проекту. Цей початковий додаток на кореновому рівні є додатком за замовчуванням для команд CLI.

3. `ng new my-workspace --create-application false` – якщо розробники збираються мати декілька проектів у одній робочій області, вони можуть пропустити початкове створення проектів під час створення робочої області та надати робочій області унікальне ім'я. Наступна команда створює робочу область із усіма файлами конфігурації для всієї робочої області, але жодної програми на кореновому рівні. Потім можна генерувати програми та бібліотеки з унікальними іменами в робочій області.

4. `cd my-workspace` – виконує перехід у папку за вказаним шляхом.

5. `ng generate library my-lib` – використовується для генерування бібліотеки, після чого новостворені файли переходять у `projects /` папку робочої

області. Бібліотеки, на відміну від програм та пов'язаних з ними проектів є2е, мають власний package.json файл конфігурації.

Модулі JavaScript та NgModules можуть допомогти розробнику зробити код модульним, проте між ними є суттєва різниця. Angular-проекти працюють з кожним з цих типами модулів.

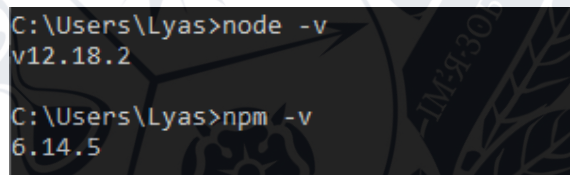
Модуль JavaScript – це окремий файл із кодом JavaScript, який зазвичай містить клас або бібліотеку функцій для використання їх з певною метою у своєму проекті. Модулі JavaScript дозволяють розподілити роботу, виконуючи її в декількох файлах для кращої читабельності та розуміння коду, дотримуючись правил написання [19].

Основна відмінність NgModules від розглянутого вже модуля JavaScript, є те, що вони не мають окремого власного файлу, а також для їх використання необхідна наявність декоратора @NgModule та його метаданих. NgModule має перелік всіх оголошених змінних та класів, а також обмежує використання класів за допомогою їх декларації. Метадані відіграють важливу роль під час компіляції в Angular, яка перетворює код програми в високопродуктивний код JavaScript. Метадані описують, як скомпілювати шаблон компонента та як створити інжектор під час виконання. Замість того, щоб визначати всі класи в одному файлі як це необхідно робити при використанні модулів JavaScript, можна оголосити які компоненти, директиви та канали належать до NgModule у списку. Ці класи називаються декларованими. NgModule може експортувати лише декларовані класи, якими він володіє, або імпортувати з інших NgModules. Він не декларує та не експортує будь-який інший клас [20].

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНТЕРАКТИВНОГО ГОЛОСУВАННЯ В ГРУПАХ ДЛЯ ЕФЕКТИВНОЇ КОМУНІКАЦІЇ НА ЛЕКЦІЙНИХ ЗАНЯТТЯХ У ВИЩИХ НАВЧАЛЬНИХ ЗАЛАДАХ

3.1. Ініціалізація проекту

Для розробки сучасного веб-додатку за допомогою мови програмування JavaScript необхідно встановити Node.js – це середовище виконання, що включає в себе всі необхідні файли для запуску програми написаної на мові програмування JavaScript [21]. Для встановлення Node.js необхідно в веб-браузері перейти на офіційний сайт та вибрати інсталятор для наявної операційної системи. Після завершення завантаження інсталятора, необхідно встановити файл на персональному комп'ютері, а потім відкрити командний рядок та написати `node -v` та `npm -v` система відобразить версію Node.js та npm (рис. 3.1).

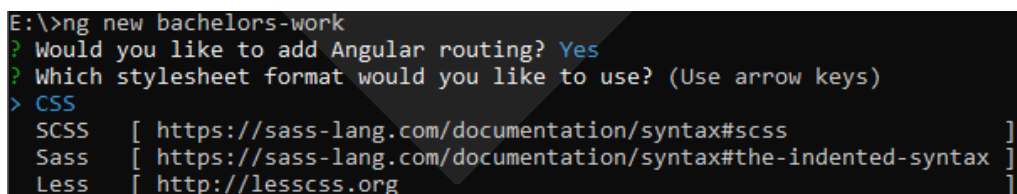


```
C:\Users\Lyas>node -v
v12.18.2

C:\Users\Lyas>npm -v
6.14.5
```

Рисунок 3.1 – Команда перевірки версії Node.js та NPM

Після встановлення Node.js можна переходити до встановлення Angular CLI. Angular CLI встановлюється командою `npm install -g @angular/cli`. За допомогою цієї команди встановлюється актуальна версія Angular. Для створення Angular-проекту в командному рядку потрібно ввести команду `ng new bachelors-work`, за нею слідують уточнюючі питання про те чи буде використовуватись Angular routing та яку каскадну таблицю стилів використовувати (рис. 3.2.) [22].



```
E:\>ng new bachelors-work
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
SCSS [ https://sass-lang.com/documentation/syntax#scss ]
Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
Less [ http://lesscss.org ]
```

Рисунок 3.2 – Створення нового Angular-проекту

В результаті виконання цієї команди створюється порожній Angular проект. Для створення необхідних для роботи компонентів використовується команда `ng generate component header`, для виконання цієї команди необхідно перейти в створену папку Angular-проекту і з неї викликати дану команду. Результатом виконання команди буде створення компоненту з назвою `header` в якому буде реалізовуватись верхня частина веб-сторінки (рис. 3.3).

```
E:\bachelors-work>ng generate component header
```

Рисунок 3.3 – Команда створення компонентів проекту

Запуск проекту відбувається за допомогою команд `npm start` або `ng serve` - о для запуску проекту (рис. 3.4).

```
E:\bachelors-work>npm start
> bachelors-work@0.0.0 start E:\bachelors-work
> ng serve
```

Рисунок 3.4 – Запуск проекту на локальному сервері

Після успішного запуску проекту, результат виконання цієї команди, веб-сторінку можна переглянути в веб-браузері вписавши в адресний рядок `http://localhost:4200/`.

3.2. UML-діаграми програмного продукту

UML-діаграма – це діаграма створена уніфікованою мовою моделювання з метою візуального представлення системи для кращого її розуміння. UML – це один з найпопулярніших методів моделювання процесів. Він базується на схематичному зображенні програмних компонентів. Використовуючи візуальні дані можна краще зрозуміти можливі недоліки або помилки програмного забезпечення. Переважно, UML використовується як мова загального призначення для моделювання в галузі програмної інженерії. Наприклад, такий тип UML-діаграм як, діаграми діяльності, можуть замінити блок-схеми. Вони забезпечують більш стандартизований спосіб моделювання робочих процесів та надають широкий спектр функцій для поліпшення читабельності та ефективності [24].

Для візуалізації роботи розробленого програмного продукту було створено діаграми прецедентів, активностей та послідовностей.

3.3. Реалізація функціоналу для ефективної комунікації на лекційних заняттях

Даний додаток реалізовано за допомогою мови програмування JavaScript, фреймворків Angular та Bootstrap і JavaScript-бібліотеки Chart.js. За допомогою даних інструментів можна створити адаптивний веб-сайт з можливістю відображення діаграм на сторінці для візуалізації результатів голосування на будь-якому девайсі.

За допомогою Angular реалізовано структуру проекту, за допомогою Bootstrap зроблена розмітка та адаптивність, а chart.js відповідає за побудову та відображення діаграм на сторінці. При переході на сайт, користувач бачить головну сторінку з переліком його університетських предметів, при відвідуванні кожного з них, студент може у режимі реального часу залишати свою оцінку стосовно якості викладеного матеріалу під час проведення лекційних занять (рис. 3.5).

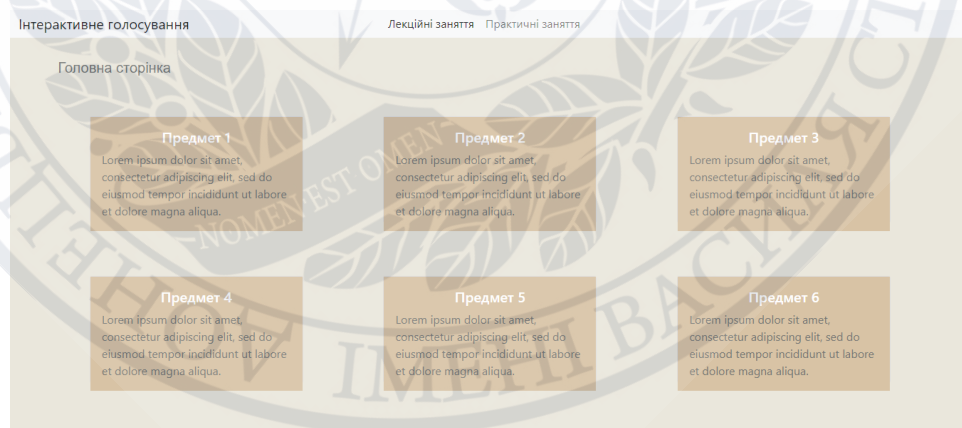


Рисунок 3.5 – Візуалізація головної сторінки програмного продукту

При переході на сторінку, обраного предмету, відкривається вікно, в якому користувач бачить перелік питань та відповідну діаграму до кожного з них, що дає можливість проводити наступні маніпуляції (рис. 3.6), а також для зручності використання під кожним з питань розміщено кнопку відправлення результатів, що дає можливість миттєво побачити зміну на графіку (Додаток А).

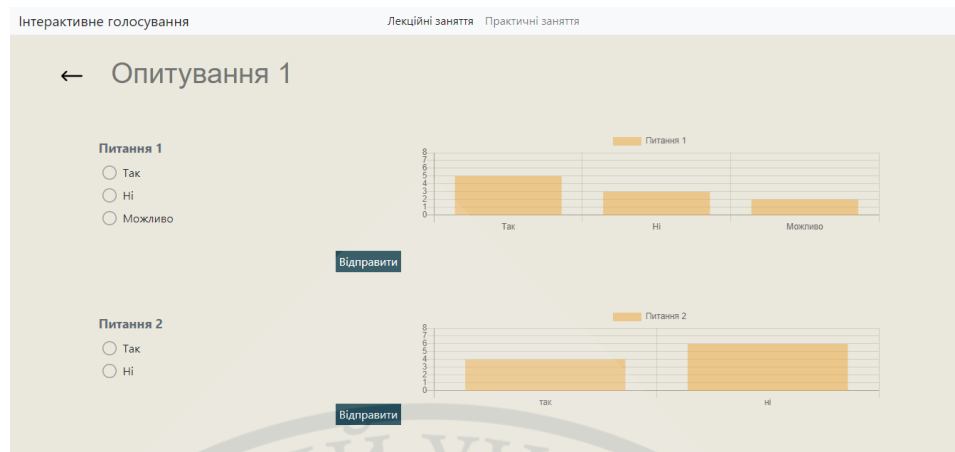


Рисунок 3.6 – Реалізація голосування на сторінці обраного предмету

Оскільки, в даній роботі немає реалізованої серверної частини, користувач виступає в ролі останнього голосуючого і в залежності від вибору варіанту відповіді буде показаний результат голосування. Для переходу між різними компонентами використовуються router link, за допомогою цього інструменту користувач не очікує на завантаження іншої сторінки після переходу, а одразу бачить контент для цієї сторінки; така швидкість переходу можлива через те, що всі сторінки завантажуються одночасно з головною сторінкою і після переходу між сторінками контент замінюється миттєво на необхідний в цьому компоненті (Додаток Б).

Питання та варіанти відповідей до конкретних сторінок прописуються в окремому файлі з розширення . json (Додаток В). В даному файлі з питаннями наведено ім'я сторінки для конкретного методу, заголовок для голосування та варіанти відповідей, назва сторінки з заголовками опитувань та варіантами відповідей для них пов'язуються за допомогою id. id – це унікальний номер який відповідає окремому елементу. Наприклад, у назви сторінки id: 1 за допомогою ідентифікатора subjectId: 1 програма розуміє, що до сторінки назва якої має id 1 відноситься питання з номером subjectId: 1, для правильного відображення варіантів відповідей до цього питання, йому також необхідно задати унікальний id. В даному випадку унікальний номер для питання – це id 1, для того щоб варіанти відповідей відображались під цим питанням до кожного з варіантів задається questionId: 1, а для визначення який варіант був вибраний у кожного

питання також є свій унікальний номер. За допомогою такого підходу, програма розуміє, що до однієї сторінки відноситься певна кількість питань з визначеними варіантами відповіді до них.

3.4. Реалізація функціоналу інтерактивного групового голосування за методом Кондорсе та Борда

В даному додатку реалізований приклад голосування методом Кондорсе та Борда. Даний функціонал дає можливість полегшити проведення групових голосувань для студентів під час практичних занять. Як приклад, можна розглянути вибір старости в групі. Студентів ділять на 5 груп з умовною кількістю студентів в кожній з них 5, 12, 7, 9, 17 і кожній з груп надають форму голосування. Форма голосування умовно складається з трьох кандидатів умовно названих А, В та С, нехай кожна група проголосує наступним чином.

Таблиця 3.1 – Розподіл голосів.

5 осіб	$A > B > C$
12 осіб	$C > A > B$
7 осіб	$B > C > A$
9 осіб	$A > C > B$
17 осіб	$C > B > A$

Порівняємо уподобання в парах кандидатів. Беремо А та С: тоді

А / С віддають перевагу $5 + 9 = 14$;

С / А віддають перевагу $12 + 7 + 17 = 36$.

Отже, С переважає за А ($C \rightarrow A$) за волею більшості.

Далі порівняємо інші пари. Беремо А та В:

А / В віддають перевагу $5 + 12 + 9 = 26$;

В / А віддають перевагу $7 + 17 = 24$.

Отже, А переважає за В ($A \rightarrow B$) за волею більшості.

Далі порівняємо інші пари. Беремо В та С:

В / С віддають перевагу $5 + 7 = 12$;

С / В віддають перевагу $12 + 9 + 17 = 38$.

Отже, С переважає за В ($C \rightarrow V$) за волею більшості.

Виходячи з результатів голосування переможцем буде кандидат С.

Зокрема у програмному продукті дані обрахунки візуалізовано наступним чином (рис. 3.7-3.8):

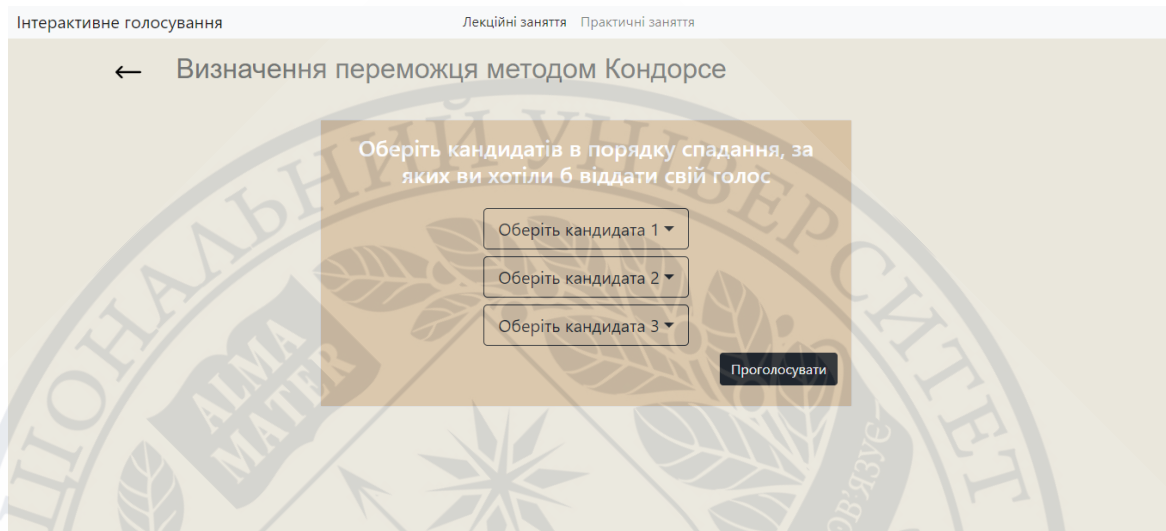


Рисунок 3.7 – Візуалізація групового голосування методом Кондорсе



Рисунок 3.8 – Результати голосування методом Кондорсе

Згідно методу Борда результати голосування виражаються у вигляді кількості балів, набраних кожним з кандидатів. Нехай кількість кандидатів дорівнює n . Тоді за перше місце присуджується n балів, за друге – $n-1$, за останнє – 1 бал.

Застосуємо метод Борда до наведеного вище прикладу. Підрахуємо кількість балів для кожного з кандидатів:

$$A = 5 \times 3 + 12 \times 2 + 7 \times 1 + 9 \times 3 + 17 \times 1 = 90$$

$$B = 5 \times 2 + 12 \times 1 + 7 \times 3 + 9 \times 1 + 17 \times 2 = 86$$

$$C = 5 \times 1 + 12 \times 3 + 7 \times 2 + 9 \times 2 + 17 \times 3 = 124$$

У відповідності з методом Борда перевагу надають так само кандидату С.

Зокрема у програмному продукті дані обрахунки візуалізовано наступним чином (рис. 3.9-3.10):

Рисунок 3.9 – Візуалізація групового голосування методом Борда



Рисунок 3.10 – Результати голосування за методом Борда

Реалізація додатку для проведення голосувань допоможе зменшити час на підрахування переможців за декількома методами голосувань, так як всі обрахування буде виконувати комп'ютер, це забезпечить швидкий та точний результат обрахунку та надасть можливість для розширення функціоналу в майбутньому.



ВИСНОВКИ

Було визначено мету бакалаврської кваліфікаційної роботи, яка *роботи* полягає у розробці інтернет ресурсу для запровадження системи організації інтерактивного голосування серед студентів задля покращення якості викладання матеріалу викладачами та миттєвої взаємодії між студентом та лектором.

Згідно поставленої мети, необхідно вирішити наступні дослідницькі завдання, серед яких: розглянуто системи голосування, види, методи та використання кожного з них, переваги та недоліки групового голосування проти досягнення консенсусу, електронні системи голосування та їх різновидності, якими користуються сьогодні; засоби реалізації застосунку, визначено мову програмування, фреймворк та бібліотеки, що використовувались в проекті, архітектуру та обрано інтегроване середовище розробки. Реалізовано програмне забезпечення, веб-ресурс, для проведення інтерактивного голосування для студентів для покращення викладання та наповнення теоретичного матеріалу викладачами, а також розширено функціонал даного веб-ресурсу з можливістю проведення інтерактивного групового голосування, що базується на методі Кондорсе та Борда.

У першому розділі було зосереджено увагу на теоретичній базі групового голосування, яка включає в себе теорію голосування та різновидність існуючих систем голосування, найвідоміші методи голосування, до яких належать голосування простої більшості, метод Кондорсе, парадокс Борда; розглянуто поняття схвального голосування, методи пошуку консенсусу, а також наведено переваги та недоліки голосування проти досягнення консенсусу в групі, які перешкоди можуть виникати за використання кожного з них, наведено типологію електронних систем голосування, які використовуються сьогодні.

У другому розділі розглянуто вибір мови програмування для створення проекту, обрано найвідповідніше по функціональним можливостям інтегрованих середовищ розробки, розглянуто фреймворк Angular2+ та його переваги для

реалізації даного програмного продукту; проведено огляд архітектури проекту, зокрема, розглянуто мову програмування TypeScript, на якій працює і побудований даний фреймворк, яка під час збирання проекту компілюється в JavaScript.

У третьому розділі наведено компоненти та програмні рішення реалізації функціоналу для ефективної комунікації на лекційних заняттях у вигляді веб-ресурсу та можливість інтерактивного групового голосування за методом Кондорсе та Борда.

Реалізовано веб-ресурс, за допомогою якого можна підвищити і покращити якісь наповнення лекційного матеріалу за рахунок швидкої комунікації між викладачами та студентами, а також наведено приклад інтерактивного голосування методами Борда та Кондорсе, яке можна використовувати в освітньому процесі під час практичних занять та для вирішення справ студентів, що стосуються освітнього процесу, де необхідно швидко вирішити певні питання шляхом групового голосування.

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

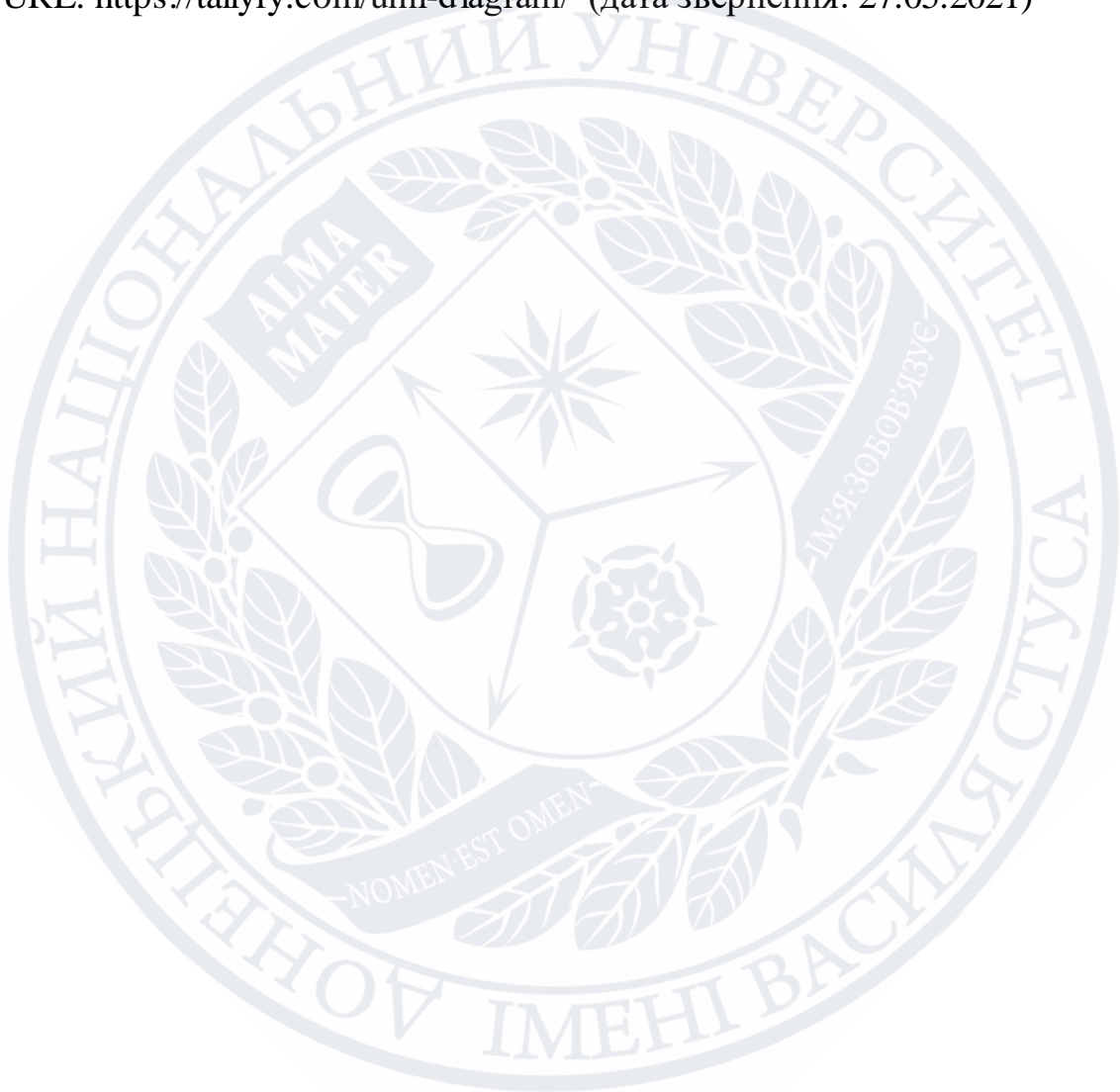
1. NE OO H. A Survey of Different Electronic Voting Systems / H. NE OO, A. AUNG. // A Survey of Different Electronic Voting Systems International Journal of Scientific Engineering and Technology Research. – 2014. – №16. – С. 3460–3464.
2. Cheng K. Voting Methods and Information Exchange in Group Support Systems / K. Cheng, F. Deek. – Acapulco, México: AMCIS, 2006. – 15 с.
3. Nurmi H. Voting Systems for Social Choice / Hannu Nurmi // Introduction to the Handbook of Group Decision and Negotiation / Hannu Nurmi., 2010. – С. 167–182.
4. Voting Methods. Stanford Encyclopedia of Philosophy URL: <https://plato.stanford.edu/entries/voting-methods/#CondPara> (дата звернення: 13.05.2021)
5. What is approval voting? URL: <https://electionscience.org/library/approval-voting/> (дата звернення: 13.05.2021)
6. Decision-Making Models: Voting versus Consensus URL: <https://healthandlearning.org/wp-content/uploads/2017/04/Decision-Making-Models-Voting-versus-Consensus.pdf> (дата звернення: 15.05.2021)
7. Wolf P. Introducing electronic voting: essential considerations / P. Wolf, R. Nackerdien, D. Tuccinardi. – Stockholm: International Institute for Democracy and Electoral Assistance, 2011. – 39 с.
8. Common Electronic Voting and Counting Technologies URL: <https://www.ndi.org/e-voting-guide/common-electronic-voting-and-counting-technologies> (дата звернення: 15.05.2021)
9. What is a Single Page Application? URL: <https://huspi.com/blog-open/definitive-guide-to-spa-why-do-we-need-single-page-applications> (дата звернення: 20.05.2021)
10. Reasons Why JavaScript is Omnipresent in Modern Development: URL: <https://snipcart.com/blog/why-javascript-benefits> (дата звернення: 20.05.2021)

11. Why did we build Visual Studio Code?: URL: <https://code.visualstudio.com/docs/editor/whyvscode> (дата звернення: 21.05.2021)
12. Why should front-end developers try WebStorm?: URL: <https://www.thinslices.com/blog/front-end-developers-webstorm> (дата звернення: 21.05.2021)
13. The Sublime Text Code Editor – An In-Depth Review: URL: <https://www.elegantthemes.com/blog/resources/the-sublime-text-code-editor-an-in-depth-review> (дата звернення: 21.05.2021)
14. The Good and the Bad of Angular Development URL: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/> (дата звернення: 23.05.2021)
15. Angular Architecture Patterns and Best Practices: URL: <https://dev-academy.com/angular-architecture-best-practices/> (дата звернення: 23.05.2021)
16. Architecture in Angular projects: URL: <https://medium.com/@cyrilletuzi/architecture-in-angular-projects-242606567e40> (дата звернення: 23.05.2021)
17. What is the difference between Javascript Module & Angular Module?: URL: <https://www.c-sharpcorner.com/interview-question/what-is-the-difference-between-javascript-module-amp-angular-module> (дата звернення: 25.05.2021)
18. Angular: Why TypeScript?: URL: <https://vsavkin.com/writing-angular-2-in-typescript-1fa77c78d8e8> (дата звернення: 25.05.2021)
19. JavaScript Modules vs. NgModules: URL: <https://nishugoel.medium.com/javascript-modules-vs-ngmodules-3da6552f0633> (дата звернення: 25.05.2021)
20. What is the difference between Javascript Module & Angular Module?: URL: <https://www.c-sharpcorner.com/interview-question/what-is-the-difference-between-javascript-module-amp-angular-module> (дата звернення: 26.05.2021)
21. About Node.js: URL: <https://nodejs.org/en/about/> (дата звернення: 26.05.2021)

22. 5 Tips & Best Practices to Organize your Angular Project: URL: <https://medium.com/dev-jam/5-tips-best-practices-to-organize-your-angular-project-e900db08702e> (дата звернення: 26.05.2021)

23. Introduction to Angular concepts: URL: <https://angular.io/guide/architecture> (дата звернення: 27.05.2021)

24. All You Need to Know About UML Diagrams: Types and 5+ Examples
URL: <https://tallyfy.com/uml-diagram/> (дата звернення: 27.05.2021)



ДОДАТОК А

Програмний код HTML-файлів

```

<nav class="navbar navbar-expand-lg navbar-light bg-light pt-0 pb-0">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Інтерактивне голосування</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-
bs-target="#navbarNav"
    aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="nav-text collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page"
routerLink="/home">Лекційні заняття</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Практичні заняття</a>
        </li>
      </ul>
    </div>
  </div>
</nav>

<div class="container">
  <div class="row pb-3">
    <div class="col">
      <h1 class="topTitle">
        Головна сторінка
      </h1>
    </div>
  </div>
  <div class="cards row pt-3 pb-5">
    <div class="col-12 col-sm-12 col-xs-12 col-md-6 col-lg-4 col-xl-4 pt-3 d-flex
justify-content-center">
      <div class="card" routerLink="/subject1">
        <div class="card-body">
          <h5 class="card-title">
            Предмет 1
          </h5>
          <p class="card-text">

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

</p>

</div>

</div>

</div>

<div class="col-12 col-sm-12 col-xs-12 col-md-6 col-lg-4 col-xl-4 pt-3 d-flex justify-content-center">

<div class="card" routerLink="/subject2">

<div class="card-body">

<h5 class="card-title">

Предмет 2

</h5>

<p class="card-text">

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

</p>

</div>

</div>

</div>

<div class="col-12 col-sm-12 col-xs-12 col-md-6 col-lg-4 col-xl-4 pt-3 d-flex justify-content-center">

<div class="card" routerLink="/subject3">

<div class="card-body">

<h5 class="card-title">

Предмет 3

</h5>

<p class="card-text">

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

</p>

</div>

</div>

</div>

<div class="col-12 col-sm-12 col-xs-12 col-md-6 col-lg-4 col-xl-4 pt-3 d-flex justify-content-center">

<div class="card" routerLink="/subject4">

<div class="card-body">

<h5 class="card-title">

Предмет 4

</h5>

<p class="card-text">

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.


```

    </p>
  </div>
</div>
</div>
<div class="col-12 col-sm-12 col-xs-12 col-md-6 col-lg-4 col-xl-4 pt-3 d-flex
justify-content-center">
  <div class="card">
    <div class="card-body">
      <h5 class="card-title">
        Предмет 5
      </h5>
      <p class="card-text">
        Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
        eiusmod tempor incididunt ut labore et dolore magna aliqua.
      </p>
    </div>
  </div>
</div>
<div class="col-12 col-sm-12 col-xs-12 col-md-6 col-lg-4 col-xl-4 pt-3 d-flex
justify-content-center">
  <div class="card">
    <div class="card-body">
      <h5 class="card-title">
        Предмет 6
      </h5>
      <p class="card-text">
        Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
        eiusmod tempor incididunt ut labore et dolore magna aliqua.
      </p>
    </div>
  </div>
</div>
</div>
</div>
</div>
<div class="container-fluid subjectPage pb-5">
  <div class="row pb-3">
    <div class="col d-flex flex-row align-items-center">
      
      <h1 class="topTitlePage">
        {{ subject1Response.name }}
      </h1>
    </div>
  </div>

```

```

</div>

<div class="row d-flex justify-content-center pt-3 pb-3">
  <div class="leftCont col-12 col-sm-12 col-xs-12 col-md-4 col-lg-4 col-xl-4 pt-
3">
    <div class="questionCont">
      <h4 class="questionTop">
        Питання 1
      </h4>
      <div class="answerCont">
        <label for="f-option" class="l-radio">
          <input type="radio" id="f-option" name="selector" value="1"
[(ngModel)]="firstSelectorTrue"
          #firstYesAnswer="ngModel">
            <span>Так</span>
        </label>
        <label for="s-option" class="l-radio">
          <input type="radio" id="s-option" name="selector" value="0"
[(ngModel)]="firstSelectorFalse"
          #firstNoAnswer="ngModel">
            <span>Hi</span>
        </label>
        <label for="s-option" class="l-radio">
          <input type="radio" id="s-option" name="selector" value="0"
[(ngModel)]="firstSelectorFalse"
          #firstNoAnswer="ngModel">
            <span>Можливо</span>
        </label>
      </div>
      <button class="pa answerBtn"
(click)="saveQuestion()">Відправити</button>
    </div>
  </div>
  <div class="rightCont col-12 col-sm-12 col-xs-12 col-md-8 col-lg-6 col-xl-6 pt-
3">
    <chart [type]="type" [data]="data" [options]="options"></chart>
  </div>
</div>

<div class="row d-flex justify-content-center pb-3">
  <div class="leftCont col-12 col-sm-12 col-xs-12 col-md-4 col-lg-4 col-xl-4 pt-
3">
    <div class="questionCont">
      <h4 class="questionTop">
        Питання 2

```

```

</h4>
<div class="answerCont">
  <label for="f-option2" class="l-radio">
    <input type="radio" id="f-option2" name="selector2">
    <span>Так</span>
  </label>
  <label for="s-option2" class="l-radio">
    <input type="radio" id="s-option2" name="selector2">
    <span>Hi</span>
  </label>
</div>
<button class="pa answerBtn">Відправити</button>
</div>
</div>
<div class="rightCont col-12 col-sm-12 col-xs-12 col-md-8 col-lg-6 col-xl-6 pt-
3">
  <chart [type]="type2" [data]="data2" [options]="options"></chart>
</div>
</div>
</div>

```


ДОДАТОК Б

Програмний код TypeScript файлів

```

import { CommonModule } from '@angular/common';
import { HttpClient, HttpClientModule } from '@angular/common/http';
import { CUSTOM_ELEMENTS_SCHEMA, NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';
import { RouterModule, Routes } from '@angular/router';
import { NgbModule } from '@ng-bootstrap/ng-bootstrap';
import { ChartModule } from 'angular2-chartjs';
import { AppRoutingModuleModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { NavBarComponent } from './components/nav-bar/nav-bar.component';
import { HomeComponent } from './pages/home/home.component';
import { Subject1Component } from './pages/subject1/subject1.component';
import { QuestionService } from './services/questions.service';
const routes: Routes = [
  {
    path: 'home',
    component: HomeComponent
  },
  {
    path: 'subject1',
    component: Subject1Component
  },
  {
    path: '*',
    redirectTo: ''
  }
];

@NgModule({
  imports: [
    AppRoutingModuleModule,
    CommonModule,
    BrowserModule,
    FormsModule,
    RouterModule.forRoot(routes),
    NgbModule,
    ChartModule,
    HttpClientModule
  ],

```

```

declarations: [
  AppComponent,
  NavBarComponent,
  HomeComponent,
  Subject1Component,
  Subject2Component,
  Subject3Component,
  Subject4Component
],
exports: [
  RouterModule
],
providers: [
  QuestionService
],
bootstrap: [AppComponent],
schemas: [CUSTOM_ELEMENTS_SCHEMA]
})
export class AppModule { }

import { Component, OnInit } from '@angular/core';
import { Subject1Options } from 'src/app/models/subject1Options';
import { Subject1Questions } from 'src/app/models/subject1Question';
import { Subject1Response } from 'src/app/models/subject1Response';
import { QuestionService } from 'src/app/services/questions.service';
import swal from 'sweetalert2';

@Component({
  selector: 'app-subject1',
  templateUrl: './subject1.component.html',
  styleUrls: ['./subject1.component.css']
})
export class Subject1Component implements OnInit {
  questions: any[] = [];
  firstCounterTrue: number = 0;
  firstCounterFalse: number = 0;
  firstSelectorTrue: number = 1;
  firstSelectorFalse: number = 0;
  subject1Response: Subject1Response = new Subject1Response(null, null, null)
  questionsResponse: Subject1Questions = new Subject1Questions(null, null, null)
  optionsResponse: Subject1Options = new Subject1Options(null, null, null)

  constructor(private _questionService: QuestionService) { }

```

```

saveQuestion() {
  swal.fire(
    'Good job!',
    'You clicked the button!',
    'success'
  )
}

```

```

type = 'bar';
data = {
  labels: ["Так", "Ні", "Можливо"],
  datasets: [
    {
      label: "Питання 1",
      data: [5, 3, 2, 0, 8],
      borderColor: "rgba(235, 168, 58, 0.5)",
      backgroundColor: "rgba(235, 168, 58, 0.5)"
    }
  ],
};

```

```

type2 = 'bar';
data2 = {
  labels: ["так", "ні"],
  datasets: [
    {
      label: "Питання 2",
      data: [4, 6, 0, 8],
      borderColor: "rgba(235, 168, 58, 0.5)",
      backgroundColor: "rgba(235, 168, 58, 0.5)"
    }
  ]
};

```

```

options = {
  responsive: true,
  maintainAspectRatio: false
};

```

```

ngOnInit(): void {
  this._questionService.getJSON().subscribe(data => {
    this.subject1Response = data as any;
    this.questionsResponse = data.questions as any;
    this.optionsResponse = data.questions.map((x: any) => x.options);
  });
}

```



```
    });  
  }  
}  
  
import { HttpClient } from '@angular/common/http';  
import { Injectable } from '@angular/core';  
import { Observable } from 'rxjs';  
  
@Injectable()  
export class QuestionService {  
  
  constructor(private http: HttpClient) {  
    this.getJSON().subscribe(data => { });  
  }  
  
  public getJSON(): Observable<any> {  
    return this.http.get("/assets/data/subject-1.json");  
  }  
}
```

ДОДАТОК В

Програмний код JSON-файлів

```
{
  "id": 1,
  "name": "Опитування 1",
  "questions": [
    {
      "id": 1,
      "subjectId": 1,
      "name": "Питання 1",
      "options": [
        {
          "id": 1,
          "questionId": 1,
          "name": "Так"
        },
        {
          "id": 2,
          "questionId": 1,
          "name": "Ні"
        },
        {
          "id": 3,
          "questionId": 1,
          "name": "Поки не готовий дати відповідь"
        }
      ]
    },
    {
      "id": 2,
      "subjectId": 1,
      "name": "Питання 2",
      "options": [
        {
          "id": 1,
          "questionId": "1",
          "name": "Так"
        },
        {
          "id": 2,
          "questionId": "1",
          "name": "Ні"
        }
      ]
    }
  ]
}
```

Декларація щодо унікальності текстів роботи
та невикористання матеріалів інших авторів без посилань

Заблуда Микола Олександрович

Прізвище, ім'я, по батькові

Інформаційних та прикладних технологій

Факультет

113 Прикладна математика

Шифр і назва спеціальності

Прикладна математика

Освітня програма

ДЕКЛАРАЦІЯ

Усвідомлюючи свою відповідальність за надання неправдивої інформації, стверджую, що подана кваліфікаційна (бакалаврська) робота на тему «Система організації інтерактивного голосування» є написана мною особисто.

Одночасно заявляю, що ця робота:

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

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

дата

підпис здобувача