

Міністерство освіти і науки України  
Український державний університет науки і технологій

Комп'ютерних технологій і систем  
(назва факультету)

Електронні обчислювальні машини  
(повна назва кафедри)

Пояснювальна записка  
до кваліфікаційної роботи  
бакалавр  
(ступінь вищої освіти)

20.06.2022  
21.06.2022

на тему: Розробка навчального комплексу «Мінімізація булевих функцій»

за освітньою програмою Комп'ютерна інженерія

зі спеціальності: 123 Комп'ютерна інженерія  
(шифр і назва спеціальності)

Виконав: студент групи: КС19120

Керівник:

Нормоконтролер:

Консультанти:

(підпис студента)

(підпис)

(підпис)

Станіслав СКЛЯР

(Ім'я ПРІЗВИЩЕ)

ст. викладач, Павло ІВІН

(посада, Ім'я ПРІЗВИЩЕ)

ст. викладач, Володимир ДЗЮБА

(посада, Ім'я ПРІЗВИЩЕ)

(назва розділу)

(підпис)

(посада, Ім'я ПРІЗВИЩЕ)

(назва розділу)

(підпис)

(посада, Ім'я ПРІЗВИЩЕ)

(назва розділу)

(підпис)

(посада, Ім'я ПРІЗВИЩЕ)

(назва розділу)

(підпис)

(посада, Ім'я ПРІЗВИЩЕ)

Засвідчую, що у цій роботі немає запозичень з  
праць інших авторів без відповідних посилань.

Студент

(підпис)

Дніпро – 2022 рік

**Ministry of Education and Science of Ukraine**  
**Ukrainian State University of Science and Technologies**

## Computer Technologies and Systems

(faculty)

## Electronic Computers

(department)

**Explanatory Note  
to Bachelor's Thesis**  
(higher education degree)

on the topic: Design of the educational complex «Minimization of Boolean functions»

according to educational curriculum	Computer Engineering
-------------------------------------	----------------------

in the Speciality: 123 Computer Engineering

(speciality and its code)

Done by the student of the group: KC19120  Sklyar Stanislav

(name, surname)

Scientific Supervisor:

/ Sr. Lecturer, Pavlo IVIN /

(position, name, surname)

**Normative controller:**

1

**Sr. Lecturer,  
Volodimir DZUBA**

(position, name, surname)

## Supervisors

/

(Chapter title heading)

(position, name, surname)

/

(Chapter title heading)

```
(position, name, surname)
```

/

(Chapter title heading)

(position, name, surname)

4

(Chapter title heading)

(position, name, surname)

**Міністерство освіти і науки України**  
**Український державний університет науки і технологій**

**Факультет: Комп'ютерні технології і системи**

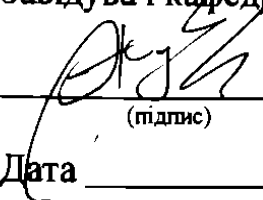
**Кафедра: ЕОМ**

**Рівень вищої освіти: Перший (бакалаврський)**

**Освітня програма: Комп'ютерна інженерія**

**Спеціальність: 123 Комп'ютерна інженерія**

(шифр та назва)

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри ЕОМ  
  
(підпис) **Ігор ЖУКОВИЦЬКИЙ**  
(Ім'я ПРІЗВИЩЕ)  
Дата \_\_\_\_\_

**З А В Д А Н Н Я**

на кваліфікаційну роботу

бакалавра

(ступінь вищої освіти)

студенту \_\_\_\_\_

**Скляру Станіславу Юрійовичу**

(Прізвище, Ім'я По батькові)

1. Тема роботи: Розробка навчального комплексу «Мінімізація булевих функцій»

Керівник роботи: \_\_\_\_\_

**Івін Павло Вікторович, б\з,б\с**

(Прізвище, Ім'я, По батькові, науковий ступінь, вчене звання)

затверджені наказом від

"07" грудня 2022р. №68ст

2. Строк подання студентом роботи: 17.06.2022 р.

3. Вихідні дані до роботи: опис методу Квайна-МакКласкі; документація до обраної технології розробки, IDE та мови програмування

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):

4.1 Постановка задачі та аналіз предметної області

4.2 Вибір програмної платформи та засобів реалізації навчального комплексу

4.3 Розробка алгоритмів та програмна реалізація навчального комплексу

4.4 Інструкція по користуванню та результати роботи навчального комплексу

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):

## 6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Завдання видав: (підпис консультанта, дата)	Завдання прийняв: (підпис студента, дата)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітки
1	Аналіз предметної області та формування постановки задачі	16.04	10%
2	Вибір програмної платформи та засобів реалізації навчального комплексу	23.04	10%
3	Розробка алгоритмів функціонування навчального комплексу	6.05	20%
4	Програмна реалізація та відладка	1.06	40%
5	Аналіз результатів та складання інструкції користувача	7.06	10%
6	Оформлення пояснювальної записки	15.06	10%
7	Подання кваліфікаційної роботи до кафедри	17.06	
8	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	22.06	


Студент

  
(підпис)

Станіслав СКЛЯР

(Ім'я ПРІЗВИЩЕ)

Керівник роботи

  
(підпис)

Павло ІВІН

(Ім'я ПРІЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи **бакалавра**:

53 сторінки, 32 рисунка, 13 джерел та 2 додатка.

**Об'єкт розробки** - навчальний комплекс, що повинен допомогти студентам самостійно освоїти один з методів мінімізації, а саме: метод Квайна Мак-Класкі.

**Мета роботи** - підвищення якості освіти, в межах викладання дисципліни «Комп'ютерна логіка» (розділ «Мінімізація булевих функцій») за рахунок використання альтернативної форми викладки матеріалу, а саме розробленого навчального комплексу.

**Одержані результати** – головним результатом роботи можна вважати web-застосунок, як програмна реалізація навчального комплексу «Мінімізація булевих функцій» та інструкція користувача, що допоможе достатньо легко інтегрувати web-застосунок до навчального процесу в межах дисципліни «Комп'ютерна логіка». Для розробки програмної частини навчального комплексу використано мову Javascript сумісно з фреймворком React. Були використані такі бібліотеки як:

- Material UI (для розробки інтерфейсів);
- Email JS (для взаємодії клієнта з email сервісами).

В демонстраційному режимі навчального комплексу реалізовано максимальну ілюстрацію та текстове супроводження всіх етапів мінімізації, з точки зору, навчальної користі пояснення матеріалу.

Тестовий режим передбачає наявність відповідного інтерфейсу реалізованому за допомогою бібліотеки Material UI, наявність автоматичної системи оцінювання, можливість зберігання відповідей та підсумкової оцінки користувача.

**Ключові слова:** МІНІМІЗАЦІЯ БУЛЕВИХ ФУНКЦІЙ, МЕТОД КВАЙНА МАК-КЛАСКІ, JAVASCRIPT, REACT, MATERIAL UI, КАЛЬКУЛЯТОР ДЛЯ МІНІМІЗАЦІЇ, EMAIL JS, МЕТОД КВАЙНА МАК-КЛАСКІ, БУЛЕВА АЛГЕБРА

## ЗМІСТ

ВСТУП .....	8
1 ПОСТАНОВКА ЗАДАЧІ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	9
1.1 Постановка задачі .....	9
1.2 Аналіз методу КМ для мінімізації булевих функцій.....	11
2 ВИБІР ПРОГРАМНОЇ ПЛАТФОРМИ ТА ЗАСОБІВ РЕАЛІЗАЦІЇ НК....	16
2.1 Програмна платформа .....	16
2.2 Середовище програмування.....	18
2.3 Мова програмування.....	19
2.4 Бібліотеки.....	20
2.5 Зберігання програмних файлів. Система контролю версій	
23_Тос92999025	
3 РОЗРОБКА АЛГОРИТМІВ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ НК .....	24
3.1 Алгоритм та програмна реалізація методу КМ .....	24_Тос92999025
3.2 Алгоритм та програмна реалізація тестового режиму НК .....	28
4 ІНСТРУКЦІЯ ПО КОРИСТУВАННЮ ТА РЕЗУЛЬТАТИ РОБОТИ НК..	32
4.1 Результат роботи демонстраційного режиму НК .....	32_Тос92999025
4.2 Результат роботи тестового режиму НК38_Тос92999025_Тос92999025	
ВИСНОВОК.....	39
ПЕРЕЛІК ПОСИЛАНЬ.....	41
ДОДАТОК А.....	43
ДОДАТОК Б .....	49

## **Перелік умовних позначень**

Метод Квайна Мак-Класкі – метод КМ

Навчальний комплекс – НК

Мінімальна диз'юнктивна нормальна форма – МДНФ

Integrated development environment – IDE

Javascript – JS

Персональний комп'ютер – ПК

## ВСТУП

Останні роки, коли дистанційна форма стала чи не єдиною можливістю отримання нових знань, вмінь та освіти в цілому, нового поштовху отримали розробки різноманітних навчальних комплексів, додатків, інтерактивних курсів тощо.

Інтегрування подібних розробок як до дистанційної так і для класичної форм навчання можуть якісно поліпшити результати навчання, а також зробити процес навчання подекуди більш зручним та зрозумілим.

В різноманітних курсах тієї чи іншої спеціальності обов'язково знайдуться розділи, певні методи і тому подібне вивчення яких можна реалізувати за допомогою програмних комплексів. Не виключенням став і курс «Комп'ютерна логіка». Так наприклад, для вивчення розділу булевої алгебри вже створені і викладені до Google Play.

При всій наявності практичного інтересу застосування вже розроблених програмних комплексів є проблема з вирішенням питання контролю знань і інтегрування їх до університетського курсу «Комп'ютерна логіка». Проблема контролю знань, головним чином, полягає в різних системах оцінювання. Тож розробка свого програмного комплексу який вирішить питання, і з боку надання знань та навичок, що цікаво студенту, а також, з боку контролю та оцінювання отриманих знань та навичок, що має робити викладач, досі є актуальною в межах конкретного курсу.

З урахуванням цього головною метою даної кваліфікаційної роботи є розробка навчального комплексу, що повинен допомогти студентам самостійно освоїти один з методів мінімізації, а саме: метод Квайна-МакКласкі.



# 1 ПОСТАНОВКА ЗАДАЧІ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Постановка задачі

Підвищення якості освіти, в межах викладання окремих дисциплін, включно і дисципліну «Комп'ютерна логіка» допустимо за рахунок використання більшої кількості різноманітних форм та сучасних засобів навчання, одними з яких є: апаратні, програмні та апаратно-програмні навчальні комплекси.

В сучасних реаліях, де дистанційна форма навчання відіграє головну роль, розробка навчальних програмних комплексів набуває особливу необхідність. Одним з розділів вивчення дисципліни «Комп'ютерна логіка» є «мінімізація булевих функцій», де студенти, головним чином, повинні придбати практичні навички застосування базових методів мінімізації.

В даній кваліфікаційній роботі передбачається розробка навчального комплексу, що повинен допомогти студентам самостійно освоїти один з методів мінімізації, а саме: метод Квайна-МакКласкі.

Відповідно до завдання в навчальному комплексі необхідно передбачити два режими роботи:

- навчально-демонстраційний;
- тестування.

**Навчально-демонстраційний режим.** В навчально-демонстраційному режимі необхідно передбачити максимальну ілюстрацію та текстове супроводження всіх етапів мінімізації, з точки зору, навчальної користі пояснення матеріалу.

Передбачається використовувати навчальний комплекс в даному режимі як при участі викладача, безпосередньо в якості допоміжного інструменту на практичних заняттях, так і дистанційно з будь якого пристрою.

**Тестовий режим.** Тестовий режим повинен передбачати наявність відповідного інтерфейсу для змоги вводу відповідей на відповідні кроки. Також цей режим повинен передбачати наявність автоматичної системи оцінювання, можливість зберігання відповідей та підсумкової оцінки користувача.

Передбачається використовувати навчальний комплекс в тестовому режимі на контрольних роботах як при участі викладача, а саме при фізичному перебуванні в аудиторії університету, так і без участі викладача, за умови дистанційної форми навчання.

Крім того в цьому режимі є необхідність показу результатів тестування користувачу, після завершення тестування. Зокрема показ правильних та неправильних відповідей і оцінка кожного кроку.

Головними задачами які необхідно вирішити в процесі виконання кваліфікаційної роботи є:

- аналіз методу мінімізації булевих функцій та визначення ключових моментів для навчально-демонстраційного та тестового режимів роботи навчального комплексу;
- вибір програмної платформи;
- вибір середовища програмування;
- вибір мови програмування, спираючись на вибрану програмну платформу;
- аналіз та вибір сторонніх сервісів (бібліотек), які допоможуть реалізувати необхідний функціонал навчального комплексу;
- вибір місця зберігання та спосіб використання електронної частини навчального комплексу;
- розробка алгоритмів для програмної реалізації НК;
- програмна реалізація;
- складання інструкції по використанню для двох режимів НК.

Також однією з головних умов є використання всіх режимів роботи з будь-якого пристрою. Наприклад для демонстраційного режиму потрібно передбачити можливість використання навчального комплексу на персональному комп'ютері. Так як викладач, щоб продемонструвати навчальний матеріал на велику аудиторію, не може обійтись без ПК та

проектору. А для тестового режиму бажано передбачити можливість використовувати цей режим на смартфоні студента.

## **1.2 Аналіз методу КМ для мінімізації булевих функцій**

Згідно з [1], логічна функція відображає роботу реального технічного пристрою. Складність функції деякою мірою визначає складність пристрою або складність програми для логічного програмованого контролера. Рівносіллям логічним функціям відповідають схеми, що виконують однакові завдання. Отже, однакові завдання можна виконувати за схемами різної складності. Тому виникає прагнення у результаті рівносілляних перетворень так видозмінити функцію, щоб отримати найпростішу схему, саме в цьому і полягає базова задача мінімізації булевих функцій.

Існують багато методів мінімізації булевих функцій:

- метод Блейка-Порецького;
- метод Нельсона;
- метод Дужкових форм;
- метод карт Карно;
- метод Квайна Мак Класкі.

В роботі передбачається розробка НК, що допоможе студентам самостійно освоїти один з методів мінімізації, а саме: метод Квайна-МакКласкі (КМ).

Вибір даного методу обумовлений різними чинниками:

- даний метод є одним з основних методів, що включенні до курсу «Комп'ютерна логіка»;
- можливість достатньо простої алгоритмічної реалізації методу.

Стислий опис методу був взятий з [2]. Алгоритм КМ, також відомий як метод простих імплікант, є методом, що використовується для мінімізації булевих функцій, який був розроблений Віллардом В. Квайном у 1952 році і розширений Едвардом Дж. МакКласкі. у 1956 році.

Алгоритм КМ функціонально ідентичний відображенню Карно, але таблицна форма робить його більш ефективним для використання в

комп'ютерних алгоритмах, а також дає детермінований спосіб перевірити, чи досягнута мінімальна форма булевої функції. Іноді його називають методом таблиць. Метод КМ є одним із найпотужніших методів мінімізації логічних виразів. У порівнянні з іншими методами, метод КМ більше виконуваний і може обробляти більше змінних.

Метод включає два головних етапи:

- знаходження всіх простих імплікант функції;
- використовуйте ці прості імпліканти в діаграмі простих імплікантів, щоб знайти суттєві прості імпліканти функції, а також інші прості імпліканти, які необхідні для покриття функції.

Хоча алгоритм КМ більш практичний, ніж відображення Карно, коли має справу з більш ніж чотирма змінними, він також має обмежений діапазон використання, оскільки проблема, яку він вирішує, є NP-повною. Час роботи алгоритму КМ експоненціально зростає із збільшенням кількості змінних. Кількість простих імплікант можна розрахувати за формолою:

$$3^n / \sqrt{n}, \quad (1.1)$$

Наприклад для 32 змінних може бути понад  $534 \times 10^{12}$  простих імплікант. Функції з великою кількістю змінних мають бути мінімізовані за допомогою потенційно неоптимальних евристичних методів.

Другий етап алгоритму зводиться до вирішення проблеми покриття набору. NP-складні екземпляри цієї проблеми можуть зустрічатися на цьому етапі алгоритму.

Далі наведений алгоритм методу КМ, який складається з шести кроків (узагальнений опис наведений в [3]).

Відповідно до завдання на кваліфікаційну роботу, будуть мінімізуватися лише функції від чотирьох аргументів, тож подальший опис методу КМ ведеться з урахуванням наведеного обмеження.

В якості вихідних даних для виконання методу КМ в нашому випадку є таблиця істинності. Приклад таблиці істинності для функції від чотирьох змінних наведений на рисунку 1.1.

#	x	y	z	t	$f(x, y, z, t)$	#	x	y	z	t	$f(x, y, z, t)$
0	0	0	0	0	0	8	1	0	0	0	1
1	0	0	0	1	0	9	1	0	0	1	1
2	0	0	1	0	0	10	1	0	1	0	1
3	0	0	1	1	0	11	1	0	1	1	1
4	0	1	0	0	1	12	1	1	0	0	1
5	0	1	0	1	0	13	1	1	0	1	0
6	0	1	1	0	0	14	1	1	1	0	1
7	0	1	1	1	0	15	1	1	1	1	1

Рисунок 1.1 – Приклад таблиці істинності для функції з чотирьох змінних

**Ключові моменти.** В навчальному комплексі передбачити можливість обрати одиничні та нульові набори функції та відображати таблицю істинності.

**Крок 1** — Розташуйте вказані мінтерми в порядку зростання та створіть групи на основі кількості одиниць, присутніх у їхніх двійкових представленнях. Таким чином, буде щонайбільше груп « $n+1$ », якщо в булевій функції є « $n$ » логічних змінних або « $n$ » бітів у двійковому еквіваленті мінтерм (рисунок 1.2).

Кількість 1	Мінтерм	Двійкове представлення
1	m4	0100
	m8	1000
2	m9	1001
	m10	1010
	m12	1100
3	m11	1011
	m14	1110
4	m15	1111

Рисунок 1.2 – Згрупована таблиця на основі кількості одиниць в мінермах

Крок 2 – Порівняйте мінтерми, присутні в послідовних групах. Якщо є зміна лише в позиції одного біта, візьміть пару цих двох мінтерм. Помістіть цей символ «-» у іншу позицію біта, а інші біти залишити без змін. Результат наведений на рисунку 1.3.

Кількість 1	Мінтерми		Імпліканти 1-го рівня	
1	m4	0100	m(4,12)	-100*
	m8	1000	m(8,9)	100-
2	m9	1001	m(8,10)	10-0
	m10	1010	m(8,12)	1-00
	m12	1100	m(9,11)	10-1
3	m11	1011	m(10,11)	101-
	m14	1110	m(10,14)	1-10
4	m15	1111	m(12,14)	11-0
			m(11,15)	1-11
			m(14,15)	111-

Рисунок 1.3 – Результат першого комбінування мінтерм

Крок 3 – Повторюйте крок 2 з новоутвореними доданками, поки не отримаємо всі прості імпліканти (рисунок 1.4).

Кількість 1	Мінтерми		Імпліканти 1-го рівня		Імпліканти 2-го рівня	
1	m4	0100	m(4,12)	-100*	m(8,9,10,11)	10--*
	m8	1000	m(8,9)	100-	m(8,10,12,14)	1--0*
2	m9	1001	m(8,10)	10-0	m(10,11,14,15)	1-1-*
	m10	1010	m(8,12)	1-00		
	m12	1100	m(9,11)	10-1		
3	m11	1011	m(10,11)	101-		
	m14	1110	m(10,14)	1-10		
4	m15	1111	m(12,14)	11-0		
			m(11,15)	1-11		
			m(14,15)	111-		

Рисунок 1.4 – Кінцевий результат комбінування

Крок 4 – Сформулюйте таблицю основних імплікантів. Вона складається з набору рядків і стовпців. Прості імпліканти можна розмістити в рядках, а

мінімальні — у стовпцях. Розмістіть «х» у клітинках, що відповідають мінтермам, які охоплюються в кожній простій імпліканті.

Крок 5 — Знайдіть основні прості імпліканти, спостерігаючи за кожним стовпцем. Якщо мінтерм охоплюється лише однією простою імплікантою, то це істотна проста імпліканта (іншими словами – ядро). Ці істотні прості імпліканти будуть частиною спрощеної булевої функції.

Крок 6 – Зменшіть таблицю простих імплікантів, видаливши рядок кожного суттєвого простого імпліканта та стовпці, що відповідають мінтермам, які охоплюються цією істотною імплікантою. Повторіть крок 5 для таблиці знижених простих імплікантів. Припиніть цей процес, коли закінчатся всі мінтерми даної булевої функції. Результат трьох попередніх кроків наведений на рисунку 1.5.

	0100	1000	1001	1010	1011	1100	1110	1111
10xx	-	+	+	+	+	-	-	-
1xx0	-	+	-	+	-	+	+	-
1x1x	-	-	-	+	+	-	+	+
x100	+	-	-	-	-	+	-	-

Рисунок 1.5 – Таблиця покриття

## **2 ВИБІР ПРОГРАМНОЇ ПЛАТФОРМИ ТА ЗАСОБІВ РЕАЛІЗАЦІЇ НАВЧАЛЬНОГО КОМПЛЕКСУ**

### **2.1 Програмна платформа**

Спираючись на список дисциплін що входять до складу компонентів освітньої програми «Комп'ютерна інженерія», ми можемо реалізувати наш навчальний комплекс 3 способами:

- Desktop додаток;
- Веб-додаток;
- Мобільний додаток.

Відповідно до [4] десктопний додаток — це повнофункціональна програма яка розміщується на комп'ютері користувача. Десктопний додаток працює ізольовано від інших додатків і вимагає наявності оператора (людини, яка працює з програмою). Людина взаємодіє із десктопну програмою за допомогою стандартного інтерфейсу.

Спираючись на джерело [5] ми можемо сказати що десктопні додатки не вимагають для своєї роботи підключення до інтернету, мають більш високу швидкодію, їх інсталивання може залежати від використовуваної операційної системи, бо десктопний додаток вимагає встановлення на кожний комп'ютер користувача, який бажає працювати з даним додатком.

Мобільний застосунок або додаток — програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях. Багато мобільних застосунків встановлені на самому пристрої або можуть бути завантажені на нього з онлайн-магазинів мобільних застосунків, таких як App Store, Google Play, Windows Phone Store та інших, безкоштовно або за плату.

Будучи встановленим на пристрій користувача, мобільний додаток є інструментом для прямого контакту з користувачем, і оптимально підходить для частого і багаторазового використання. [6].

Веб-додаток – це програмне забезпечення або програма, яку можна відкрити за допомогою будь-якого браузера. Зовнішній інтерфейс веб програми



розробляється за допомогою таких мов програмування: HTML, CSS, Javascript, які підтримуються на будь-якому браузері (Opera, Chrome, Mozilla, Edge) У той час як для написання серверної частини (Back-end) може використовуватися будь-яка інша мова програмування або фреймворк, Python, PHP, Ruby, Java.

Після аналізу способів реалізації поставленої в кваліфікаційній роботі задачі прийшли висновку використати саме веб додаток.

Спираючись на джерело [7], далі наведені основні перевагами веб додатків.

Доступність. Програма доступна з будь-якої точки планети, де є доступ до Інтернету, з будь-якого пристрою:

- комп'ютера;
- ноутбука;
- смартфона;
- планшета і т.д.

Адаптивність. На відміну від нативних продуктів, веб додатки не вимагають конкретної операційної системи. Вони працюють на всіх ОС і у всіх існуючих браузерах:

- Opera;
- Chrome;
- Firefox;
- Edge.

З огляду на вищесказане, слід окремо відзначити їх економічність. Адже немає необхідності розробляти окремі продукти для кожної ОС або браузера.

Відсутність програмного забезпечення для клієнтів. Не потрібно нічого завантажувати і встановлювати. Це спрощує процес користування.

Також суттєво оптимізується процес обслуговування, модернізації інтерфейсу. Всі оновлення активуються автоматично при завантаженні сторінки.

На відміну від мобільних додатків, для веб додатків не потрібно схвалення жодних платформ, щоб випустити свою програму;

Користувальницькі клієнти не залежать від обладнання і не вимагають налаштувань. Процес підключення нового користувача закінчується наданням йому посилання на додаток.

Отже, спираючись на основні переваги веб додатків і спираючись на ті цілі та вимоги, які поставлено в розділі «постановка та аналіз предметної області», доцільно обрати саме веб додаток.

## **2.2 Середовище програмування**

Так як для реалізації нашого навчального комплексу був обраний саме веб додаток, то нам потрібно обрати середовище програмування (далі IDE), яке підходить саме для розробки веб додатків.

IDE — комплексне програмне рішення для розробки програмного забезпечення. Складається з редактора початкового коду, інструментів для автоматизації компілювання та відлагодження програм. Більшість сучасних середовищ розробки мають можливість автодоповнення коду.

Найпопулярніші на сьогодні IDE для розробки веб додатків:

- VS Code;
- Webstorm;
- Sublime Text;
- Atom.

Вибір було зроблено в користь - Visual Studio Code (рисунок 2.1), який розробляється компанією Microsoft. Існує багато переваг використання Visual Studio Code над будь-яким іншим IDE.

Основні переваги такі: (порядок створений на основі поставлених задач в розділі 1, від найбільш значущого до менш значущого):

- безкоштовний;
- кроссплатформенний. Працює на Windows, Linux та Mac OS;
- підтримує більшість мов програмування (Python, JavaScript, TypeScript, C++, Java, PHP);

- забезпечує мовну документацію і має потужний інструмент автодоповнення коду;
- вбудована інтеграція Git.

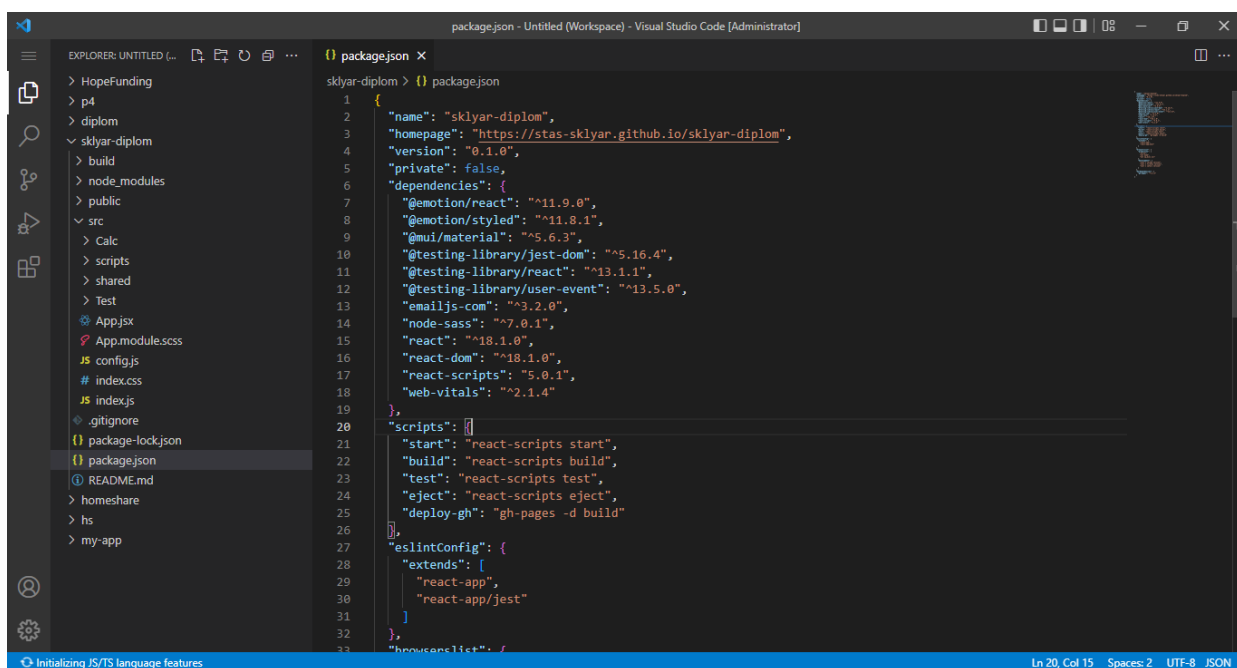


Рисунок 2.1 – Інтерфейс Visual Studio Code

## 2.3 Мова програмування

В якості мови програмування вибрано мову JavaScript, адже, в більшості браузерів за замовчування мова сценаріїв – це саме JS.

JavaScript – це мова програмування, що дозволяє зробити веб-сторінку інтерактивною, тобто такою що реагує на дії користувача.

Як описано в [8], послідовність інструкцій (що називається програмою, скриптом або сценарієм) виконується інтерпретатором, вбудованим в звичайний Web -браузер. Іншими словами, код програми вбудовується в HTML-документ і виконується на боці клієнта.

Для виконання програми не потрібно навіть перезавантажувати Web -сторінку, всі програми виконуються в відповідь на будь-яку подію. Наприклад, перед відправленням даних форми можна перевірити їх на допустимі значення і, якщо значення не відповідають очікуваним, заборонити відправлення даних.

## 2.4 Бібліотеки

Для інтерактивності веб застосунку, для більш зручної взаємодії з DOM і для зберігання всіх станів наших обчислень доцільно JS фреймворк.

Зважаючи на наші задачі, на сьогодні існують 3 найбільш популярних фреймворка:

- React;
- Angular;
- Vue.

Вибір зроблено на користь React. По-перше, згідно з [9], React це найбільш розповсюджений фреймворк із списку наведеного вище (більш детальна інформація наведена на рисунку 2.2), а також він на сто відсотків вирішує всі наші проблеми, які було поставлено в постановці задачі.

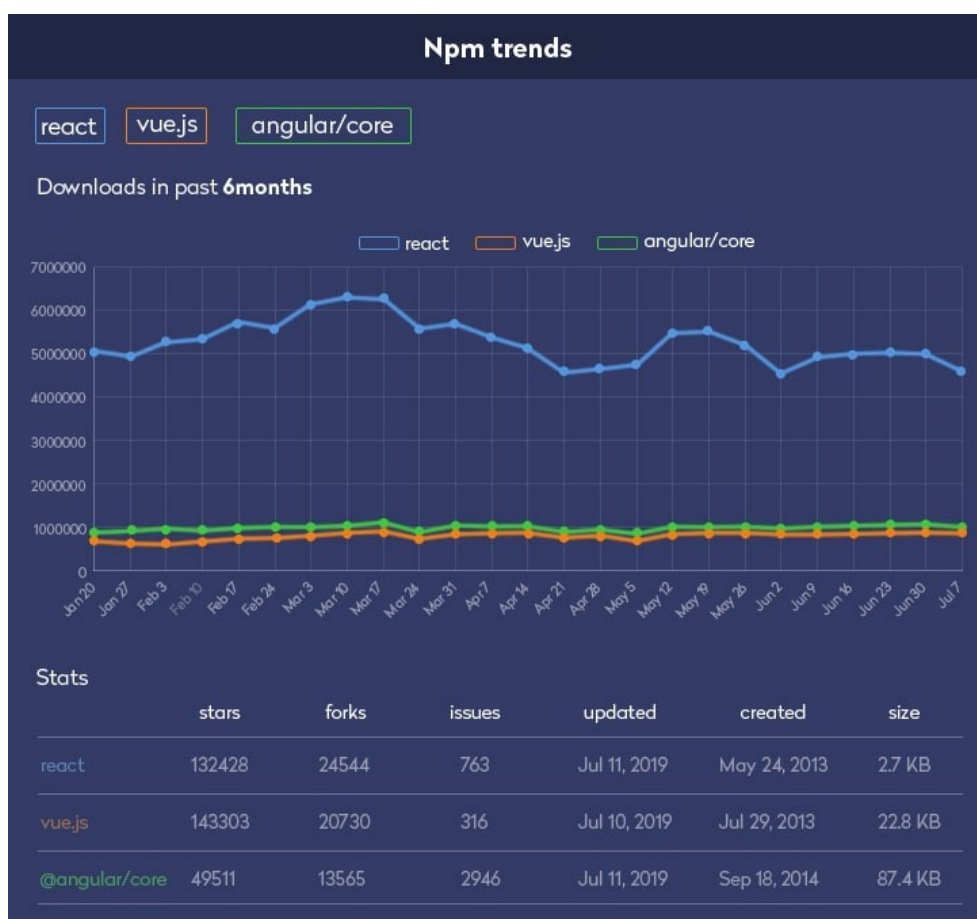


Рисунок 2.2 – Порівняльна статистика react, vue і angular

По-друге, під час вивчення дисципліни «Розробка мобільних пристроїв та додатків», для розробки проекту був використаний фреймворк - React Native. Так як в React Native принципи та методи розробки взяті саме з React, то для розробки навчального комплексу доцільно було вибрати саме React.

Згідно з інформацією з веб сайту [10], React — відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб сторінки, з якими стикаються в розробці односторінкових застосунків. Вона дозволяє компонувати складні інтерфейси з невеликих окремих частин коду — «компонентів».

Оскільки логіка компонентів написана на JavaScript, замість шаблонів, ми з легкістю можемо передавати складні дані у нашому додатку і зберігати стан окремо від DOM.

Самостійне створення дизайну для інтерфейсу нашого веб застосунку не передбачається, тож для вирішення цієї проблеми буде використано JS бібліотеку - Material UI, оскільки вона містить уже готові стилізовані компоненти, що значно скорочує час розробки комплексу в цілому.

Як описано на офіційному сайті [11], Material UI — одна з найбільш популярних бібліотек компонентів React. Компоненти та всі шаблонні елементи створені відповідно до принципів Google Material UI, але водночас вони є компонентами React.

Особливість бібліотеки Material UI в тому, що вона фокусується на фізичному світі й текстурах, але зберігає мінімалізм в елементах користувацького інтерфейсу. Приклади готових компонентів Material UI наведені на рисунку 2.3.

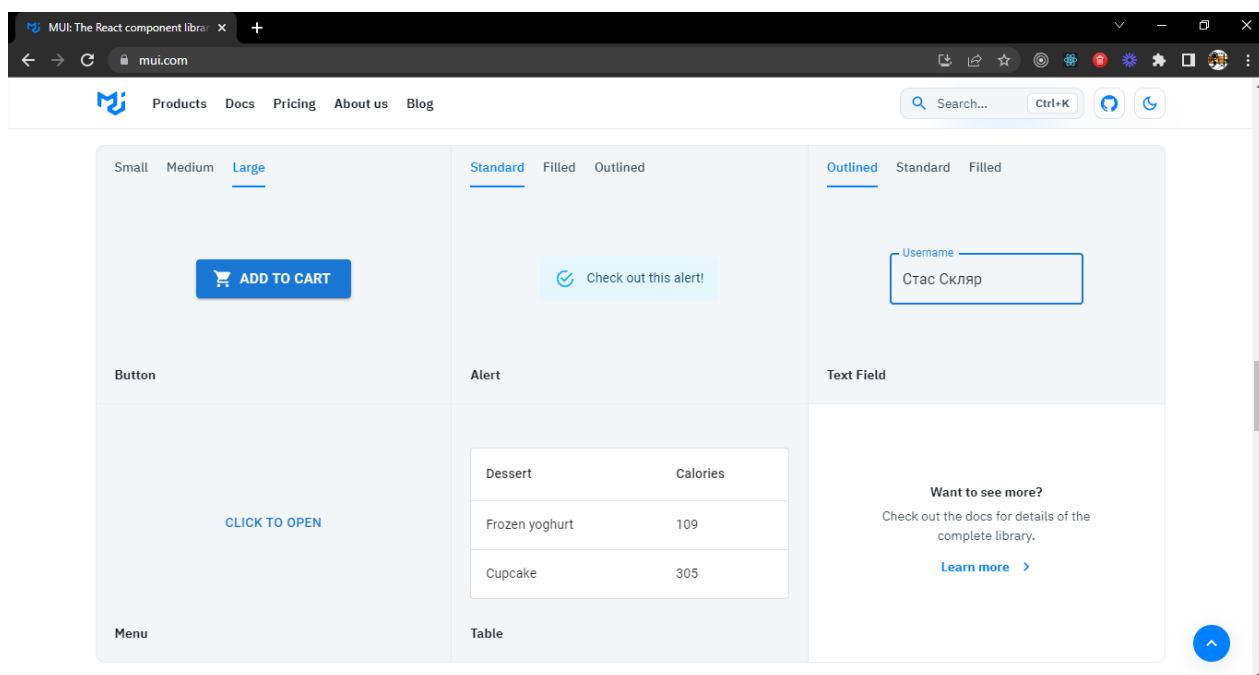


Рисунок 2.3 – Приклади компонентів Material UI

Існує багато бібліотек які вирішують проблему відправки готових шаблонів на email. В нашому додатку, для відправлення результатів тесту викладачу на пошту, ми будемо використовувати EmailJS [12], одну з найбільш популярних бібліотек для взаємодії веб додатків з email сервісами.

EmailJS допомагає надсилати електронні листи, використовуючи лише клієнтські технології. Тобто сервер для цього не потрібний.

Шаблони електронної пошти можуть додатково містити динамічні змінні практично в будь-якому полі (наприклад, тема, зміст, адреса КОМУ і т. д.), які заповнюються з Javascript. Наприклад, тема може бути "{{ name }}", пройшов тест {{ дата }}. Результат {{ result }} балів.", а за допомогою JavaScript ім'я, дата і результат може бути встановлене, наприклад, "Стас Скляр, пройшов тест 02.06.2022. Результат 77 балів.". Графічний інтерфейс бібліотеки EmailJS наведений на рисунку 2.4.

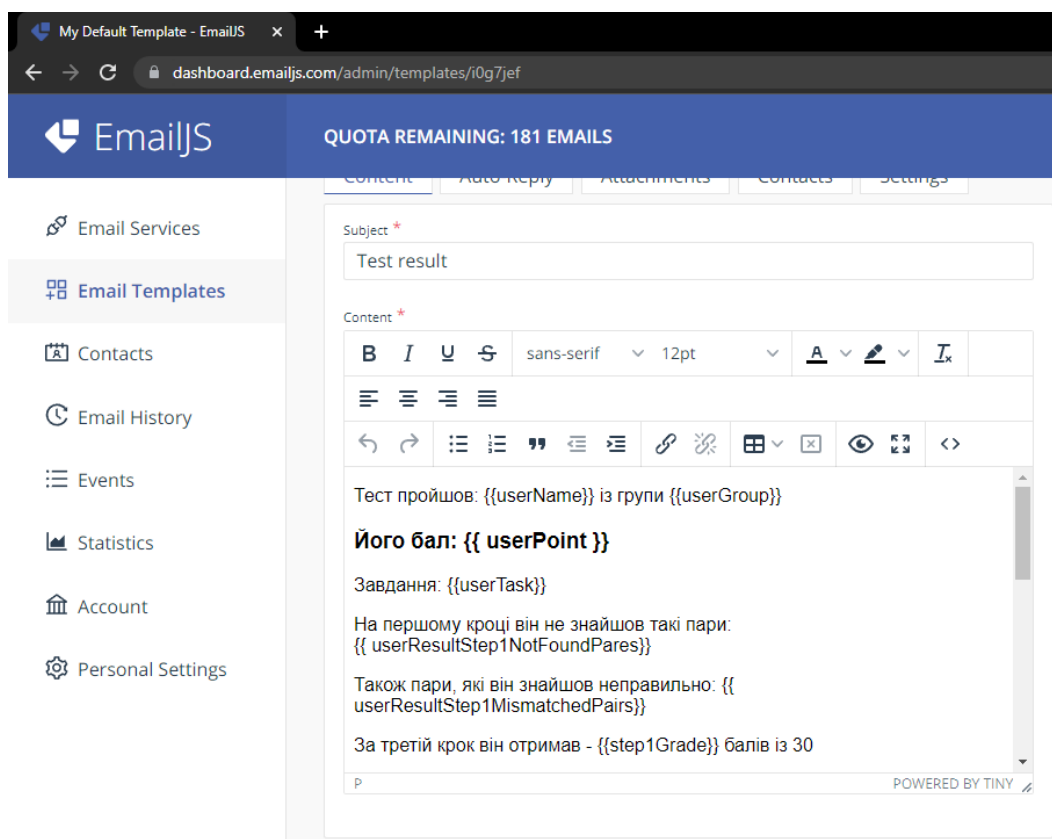


Рисунок 2.4 – Приклади компонентів Material UI

## 2.5 Зберігання програмних файлів. Система контролю версій

Існують ризики того що під час розробки файли програми зникнуть за невідомою причиною (вірус, людський фактор), а також ризики того що ми можемо зробити в нашій програмі якісь критичні зміни і в нас не буде можливості «відкотити» лише останні зміни (які роблять неможливим використання програми). Для вирішення цієї проблеми ми будемо використовувати Git і GitHub.

Git – це програмне забезпечення для забезпечення контролю версій. Github - найпопулярніше віддалене сховище для git-репозиторіїв. Git дозволяє відстежувати кожну ревізію (зміну) і за потреби є можливість «відкату» програми до попередньої версії в декілька кліків [13].

## 3 РОЗРОБКА АЛГОРИТМІВ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ НК

### 3.1 Алгоритм та програмна реалізація методу КМ

Не зважаючи на режим роботи навчального комплексу, для мінімізації булевої функції методом КМ, алгоритм який наведений на рисунку 3.1.

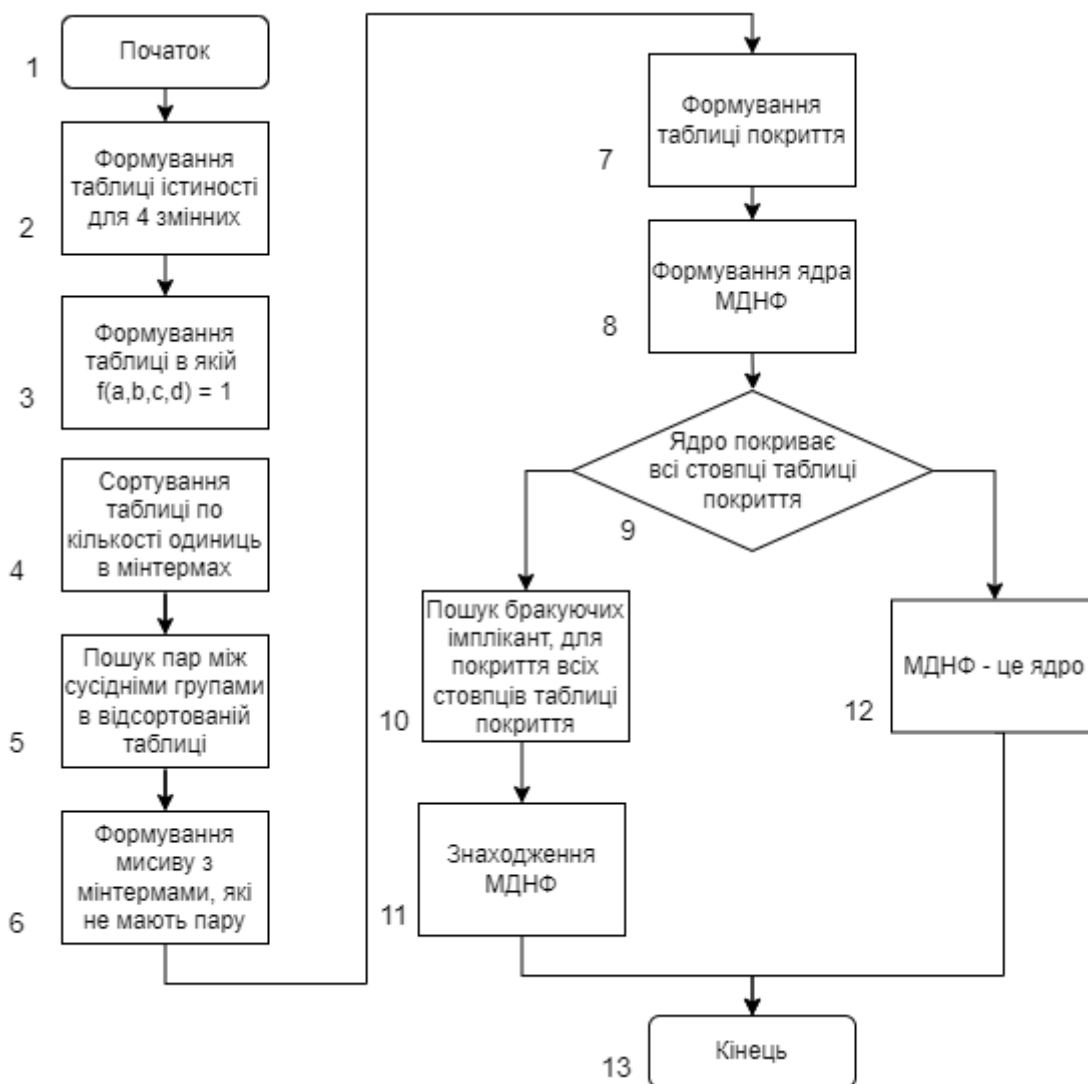


Рисунок 3.1 – Алгоритм мінімізації булевої функції методом КМ

Щоб почати етап мінімізації булевої функції методом КМ спочатку необхідно скласти таблицю істинності (блок 2, рисунок 3.1). Приклад таблиці істинності для функції з чотирма змінними наведена в розділі 1 на рисунку 1.1.

Далі потрібно сформувати нову таблицю з мінтермами для яких  $f(a,b,c,d) = 1$  (блок 3, рисунок 3.1), а також потрібно згрупувати мінтерми по кількості



одиниць в них (блок 4, рисунок 3.1). Узагальнена ілюстрація процесу формування наведена на рисунку 3.2.

#	x	y	z	t	$f(x,y,z,t)$	#	x	y	z	t	$f(x,y,z,t)$	Кількість 1	Мінтерм	Двійкове представлення
0	0	0	0	0	0	8	1	0	0	0	1	1	m4	0100
1	0	0	0	1	0	9	1	0	0	1	1		m8	1000
2	0	0	1	0	0	10	1	0	1	0	1	2	m9	1001
3	0	0	1	1	0	11	1	0	1	1	1		m10	1010
4	0	1	0	0	1	12	1	1	0	0	1		m12	1100
5	0	1	0	1	0	13	1	1	0	1	0	3	m11	1011
6	0	1	1	0	0	14	1	1	1	0	1		m14	1110
7	0	1	1	1	0	15	1	1	1	1	1	4	m15	1111

Рисунок 3.2 – Формування таблиці з мінтермами для яких  $f(a,b,c,d) = 1$

Алгоритм для формування і групування таблиці:

- запускаємо перший цикл. На кожній ітерації буде формуватися масив мінтерм сума одиниць яких буде дорівнювати поточному індексу;
- запускаємо другий цикл, який буде проходити по всім мінтермам із таблиці істинності для яких  $f(a,b,c,d) = 1$ ;
- після завершення другого циклу до масиву результату, додаємо масив мінтерм з однаковими кількостями одиниць.

Для реалізації цього алгоритму використано допоміжну функцію *sortByAmount()*, лістинг якої наведений в додатку А. В якості аргументів функції передано вихідний масив мінтерм, для яких  $f(a,b,c,d) = 1$ . В результаті буде сформований масив, елементами якого будуть масиви з однаковими кількостями одиниць (максимум 4 елементи). Приклад такої таблиці був наведений в розділі 1 на рисунку 1.2.

Далі згідно з алгоритму (блок 5, рисунок 3.1) потрібно комбінувати між собою мінтерми (іншими словами проводити операцію склеювання). Якщо два мінтерми відрізняються лише на один символом, що стоїть в той же самій позиції в обох мінтермах, заміняємо цей символ на «х», це означає, що даний символ в подальшому для нас не має значення.

При переході до імплікант другого рангу, трактуємо «х» як третє значення. Наприклад: x111 і x110 або x011x можуть бути комбіновані, а x110 і 011x не можуть.

Порядок дій для комбінування таблиці:

- в якості вихідної таблиці використовуємо масив із згрупованими і відсортированими по кількості одиниць мінтермами;
- за допомогою циклу ми проходимо по всіх елементах вихідного масива і в функцію *findPares()*, яка відповідає за знаходження пар мінтермів, передаємо елемент вихідного масива з поточним індексом і з наступним індексом. Так як за правилами, комбінувати ми можемо тільки сусідні групи.

Пошук пар для мінтерма:

- за допомогою циклу ми проходимо по всіх елементах вихідного масиву. Цим самим ми забезпечуємо можливість перевірити наявність пари для всіх мінтермів в масиві;
- як тільки отримаємо поточний мінтерм, запускаємо ще один цикл за допомогою якого ми ще раз пройдемо по вихідному масиві і порівнюємо поточну мінтерму з усіма іншими мінтермами вихідного масива;
- далі порівнюємо між собою дві мінтерми. Якщо кількість невідповідностей дорівнює одному це означає що ми знайшли пару.
- якщо пару не знайдено мінтерм додається в масив який відповідає за зберігання імплікант для яких не знайшлося пари (цей масив знадобиться на етапі формування таблиці покриття).

Фінальний результат комбінування зображений на рисунку 1.4.

Наступним етапом мінімізації є формування таблиці покриття (блок 7 рисунок 3.1). Рядками в цій таблиці виступає останній масив імплікант, який отриманий на останньому етапі комбінування, об'єднаний з масивом імплікант, для яких не було знайдено жодної пари. В якості стовпців

використовуються мінтерми отриманні на першому етапі мінімізації. Тобто мінтерми для яких  $f(a,b,c,d) = 1$ .

Далі згідно алгоритму переходимо до розставляння міток в описаній вище таблиці. Якщо перша імпліканта входить в мінтерм, ставимо позначку на перетині. Таку ж саму процедуру робимо для інших імплікант. В результаті виходить таблиця схожа на таблицю зображену на рисунку 1.5.

Далі згідно алгоритму та на основі таблиці покриття, можемо визначити ядро МДНФ (блок 8, рисунок 3.1). Ядром будуть імпліканти для яких в відповідном стовпці стоїть тільки одна мітка. В нашому випадку це імпліканти: 10xx, x1x1, x100 (на рисунку 1.5 вони позначенні жирми шрифтом).

Після того як було визначено ядро, починається етап «викреслювання стовпців» (блок 10, рисунок 3.1). Спочатку викреслюємо стовпці які мають перетини з ядром, яке ми знайшли на минулому етапі. Для стовпців які не мають перетину з ядром є можливість вибрати довільну імпліканту, але потрібно щоб ця імпліканта входила в мінтерм цього стовпця. Якщо ядро покриває всі стовпці, то відповідно ніякі імпліканти вибирати не потрібно.

Далі за допомогою логічної операції диз'юнкції, ми поєднаємо ядро й вибрані нами, на попередньому етапі, імпліканти (в наведеному прикладі МДНФ є ядром). Імпліканти для МДНФ: [10xx], [x1x1], [x100].

Останній етап полягає в перетворенні МДНФ вигляду “10xx v x1x1 v x100” в МДНФ вигляду «a!b v bc v b!c!d». Для цього перетворення використано функцію *fromNumToLetterForFourImplicant()*, аргументом якої є масив імплікант які належать до МДНФ.

За допомогою циклу «проходимо» по масиву імплікант і в залежності від відповідної позиції символу ми змінюємо його на літеру. Якщо символ дорівнює «х», то цей символ не додається в рядок результату. Якщо символ дорівнює «0», то цей символ додається в результат разом з символом логічного заперечення. Лістинг функції *fromNumToLetterForFourImplicant()* наведений в додатку А.

### 3.2 Алгоритм та програмна реалізація тестового режиму НК

Алгоритм тестового режиму роботи навчального комплексу представлений на рисунку 3.3.

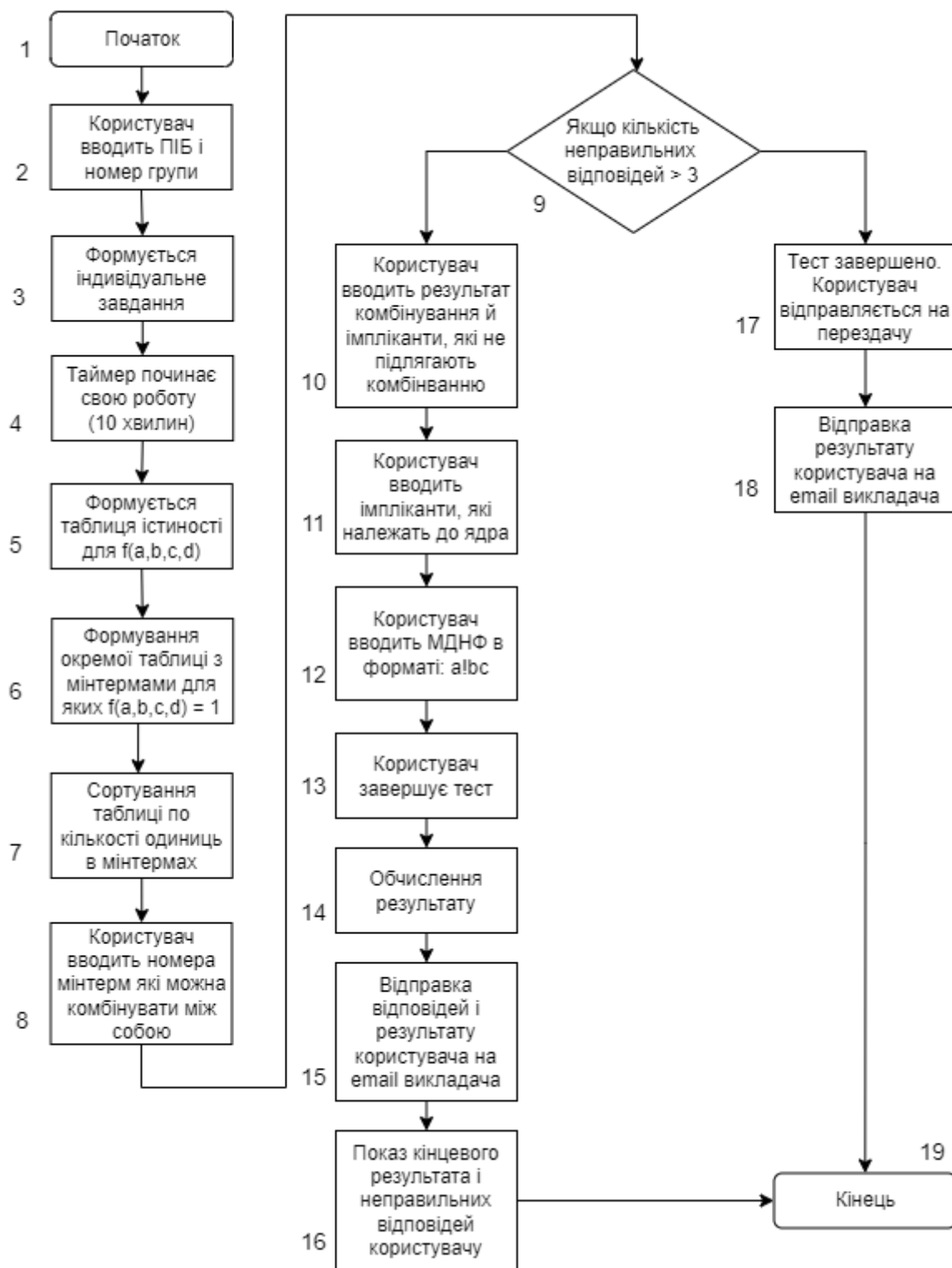


Рисунок 3.3 – Алгоритм тестового режиму роботи

На початку тестування користувач вводить своє ПІБ і номер групи. Ці поля не можуть бути пустими.

Далі користувач тисне кнопку «Почати тест». Після цього формується індивідуальне завдання для користувача і починає свою роботу таймер. На тест відводиться 40 хвилин.

На наступному кроці, реалізація якого наведена в підрозділі 3.1, формується таблиця істинності для 4 змінних, з якої обираються всі рядки з мінтермами для яких  $f(a,b,c,d) = 1$  і на їх основі формується нова таблиця (блок 5 та блок 6, рисунок 3.3). Потім цю таблицю потрібно згрупувати по кількості одиниць в мінтермах.

Наступний крок тестування (блок 8, рисунок 3.3) полягає в тому що користувач вводить номери мінтерм, які можна комбінувати між собою. Для цього в відповідній секції інтерфейс передбачає наявність текстового поля і кнопки для вводу номерів мінтерм. А також є можливість видаляти з списку вже введені імпліканти.

Якщо на цьому етапі користувач ввів більше трьох правильних відповідей, то тест для нього закінчено і він відправляється на перездачу. Користувач отримує повідомлення про невдачу спробу тестування, а викладач, за допомогою бібліотеки Email.js, отримує електронний лист на свій email, який містить ПІБ користувача, номер його групи і список неправильних відповідей користувача (приклад електронного листа зображений на рисунку 3.4).

Failed Test Скляр Станіслав Юрійович КС-19-120 Входящие x

stas &lt;stas.sklyar.0000@gmail.com&gt;

кому: мне ▾

Студент Скляр Станіслав Юрійович із групи КС-19-120 зробив більше трьох помилок на 1 кроці

Його відповіді:

1,1 | 2,2 | 3,3 | 4,4

Його завдання:

0000100011111011

Це означає, що студент відправляється на перездачу

Email sent via [EmailJS.com](https://EmailJS.com)

## Рисунок 3.4 – Приклад електронного листа після невдалої спроби

Якщо на данному етапі користувач не зробив більше трьох помилок – виконується (блок 10 та блок 11, на рисунку 3.3). Користувач вводить результат комбінування й імпліканти, які не підлягають комбінванню. І після цього потрібно вибрати і ввести імпліканти, які належать до ядра МДНФ (цієї функції).

Інтерфейс для цих двох кроків ідентичний до інтерфейсу із попереднього кроку. Тобто є текстове поле для вводу імпліканти, кнопка для того щоб додати імпліканту до списку і відповідна іконка, для того щоб видалити імпліканту із списку.

На наступному кроці, який наведений в (блок 12, рисунок 3.3), користувач повинен ввести МДНФ в форматі: «a!bc». Для цього передбачено текстове поле і набір кнопок для вводу змінних – «a», «b», «c», «d», «!», «v», а також кнопка для видалення символів – «←».

Спираючись на завдання, яке наведено в постановці задачі в розділі 1, потрібна система оцінювання результатів користувача. Система оцінки користувача розроблена таким чином, що на кожний крок тестування передбачається відповідна кількість балів. Всього можна отримати - 100 балів.

Кількість балів на кожний крок:

- перший крок - 30 балів;
- другий крок - 25 балів;
- третій крок - 25 балів;
- четвертий крок - 20 балів.

Якщо наприклад на першому кроці користувач повинен вказати 10 відповідей, то «вага» кожної відповіді дорівнює - 3. Тобто якщо користувач не вказав одну із правильних відповідей, то система віднімає 3 бали від сумарної кількості балів за відповідний крок. Якщо користувач вказав неправильну відповідь система віднімає половину «ваги» однієї відповіді. В нашому випадку система відніме 1.5 бала.

Такий самий механізм використовується і для наступних кроків. Відмінність лиш в кількості балів за відповідний крок. За програмну реалізацію процесу оцінювання користувача відповідає функція *calcUserGrade()*, яка наведена в додатку А.

Після закінчення тестування, результат користувача відправляється в електронному повідомленні на email викладача. В ньому присутня вся інформація про користувача, його остаточний бал, індивідуальне завдання, його відповіді, а також кількість балів за кожний крок. Цьому етапу відповідає блок 15 на рисунку 3.3. За програмну реалізацію процесу відправки електронного повідомлення відповідає функція *sendEmail()*.

Далі у користувача є можливість переглянути свої результати, а також свої правильні і неправильні відповіді. Також користувач може переглянути детально, яку кількість балів за який крок він отримав.

## 4 ІНСТРУКЦІЯ ПО КОРИСТУВАННЮ ТА РЕЗУЛЬТАТИ РОБОТИ НК

### 4.1 Результат роботи тестового режиму НК

Доступ до тестового режиму НК користувач отримує переходячи по посиланню, яке отримує від викладача. Після переходу за посиланням користувач опиняється на сторінці тестового режиму. На цій сторінці необхідно ввести ПІБ користувача та номер групи користувача (рисунок 4.1).

The screenshot shows a web interface for a test mode. At the top, there are two tabs: 'КАЛЬКУЛЯТОР' (Calculator) and 'ТЕСТ' (Test), with 'ТЕСТ' being the active tab. Below the tabs, there are two input fields. The first is labeled 'Введіть ПІБ' (Enter PIB) and contains the text 'Скляр Станіслав Юрійович'. The second is labeled 'Введіть групу' (Enter group) and is empty. Below these fields is a blue button labeled 'ПОЧАТИ ТЕСТ' (Start Test).

Рисунок 4.1 – Інтерфейс введення особистих даних користувача

Далі користувачу потрібно натиснути кнопку «ПОЧАТИ ТЕСТ». Після (згідно з алгоритму наведеному в розділі 3) зверху сторінки відображається таймер і формується таблиця істинності для чотирьох змінних (рисунок 4.2). Потім користувач тисне кнопку «НАСТУПНИЙ КРОК». Після натискання якої формується таблиця з мінтермами для яких  $f(a,b,c,d) = 1$  (рисунок 4.3).



КАЛЬКУЛЯТОР

ТЕСТ

Залишилось: 9 хвилин

ВАШЕ ЗАВДАННЯ: 0000100011111011

Формуємо таблицю істинності

№	a	b	c	d	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0

Рисунок 4.2 – Ілюстрація таблиці істинності та таймеру

Всі рядки з мінтермами для яких  $f(a,b,c,d) = 1$ 

№	a	b	c	d	F
0	0	1	0	0	1
1	1	0	0	0	1
2	1	0	0	1	1
3	1	0	1	0	1
4	1	0	1	1	1
5	1	1	0	0	1
6	1	1	1	0	1
7	1	1	1	1	1

НАСТУПНИЙ КРОК

Рисунок 4.3 – Таблиця з мінтермами для яких  $f(a,b,c,d) = 1$

Далі (згідно алгоритму наведеному в розділі 3) користувач повинен ввести номери мінтерм для подальшого комбінування. Номер вводиться в текстове поле, після цього потрібно натиснути кнопку «ДОДАТИ ПАРУ». Також є можливість видалення доданих номерів. Для цього потрібно навпроти рядка з номерами пар натиснути на іконку у вигляді «корзини». Після введення всіх відповідей потрібно натиснути кнопку «НАСТУПНИЙ КРОК» (рисунок 4.4).

**1. КРОК**

Введіть в поле номера імплікант які ми повинні склеяти

Введіть індекси пар імплікант

1

ДОДАТИ ПАРУ

[1 - 4],

[2 - 5]

НАСТУПНИЙ КРОК

Рисунок 4.4 – Інтерфейс для введення номерів мінтерм для подальшого комбінування

Далі користувач повинен ввести імпліканти отримані після комбінування та імпліканти які не підлягають комбінуванню. Номер вводиться в текстове поле, після цього потрібно натиснути кнопку «ДОДАТИ ІМПЛІКАНТУ». В текстовому полі є можливість введення тільки символів «1», «0» та «х». Після введення всіх відповідей потрібно натиснути кнопку «НАСТУПНИЙ КРОК» (рисунок 4.5).

Далі для введення імплікант які належать до ядра МДНФ необхідно виконати такий же порядок дій, який наведений в попередньому абзаці (рисунок 4.6).

## 2. ДРУГИЙ КРОК

Введіть отриманні імпліканти після завершення комбінування й імпліканти які **не** підлягають комбінванню.

Введіть імпліканту

1xx1|

ДОДАТИ ІМПЛІКАНТУ

[01x1],

[0xx1]

НАСТУПНИЙ КРОК

Рисунок 4.5 – Інтерфейс введення імплікант отриманих після комбінування та імплікант які не підлягають комбінуванню

## 3. ТРЕТІЙ КРОК

Введіть імпліканти які належать до ядра МДНФ в форматі: [1,0,x,1]

Введіть імпліканту

ДОДАТИ ІМПЛІКАНТУ

[1x1x],

[110x]

НАСТУПНИЙ КРОК

Рисунок 4.6 – Ілюстрація процесу введення ядра МДНФ

Після цього користувачу потрібно ввести МДНФ. Для цього потрібно використовувати кнопки «a», «b», «c», «d», «!», «v» та «←» (для видалення введеного символу). Після введення відповіді необхідно натиснути кнопку «ЗАВЕРШИТИ ТЕСТ» (рисунок 4.7).

**4. ЧЕТВЕРТИЙ КРОК**

Введіть МДНФ в форматі - ab!c v b!c

Введіть МДНФ  
 ab v !cd v a!bd

a

b

c

d

!

v

←

ЗАВЕРШИТИ ТЕСТ

Рисунок 4.7 – Інтерфейс секції введення ядра МДНФ

Після натискання кнопки з'являється модальне вікно де для підтвердження відправки тесту необхідно натиснути кнопку «ВІДПРАВИТИ», а для того щоб повернутися до тесту необхідно натиснути кнопку «НАЗАД» (рисунок 4.8).

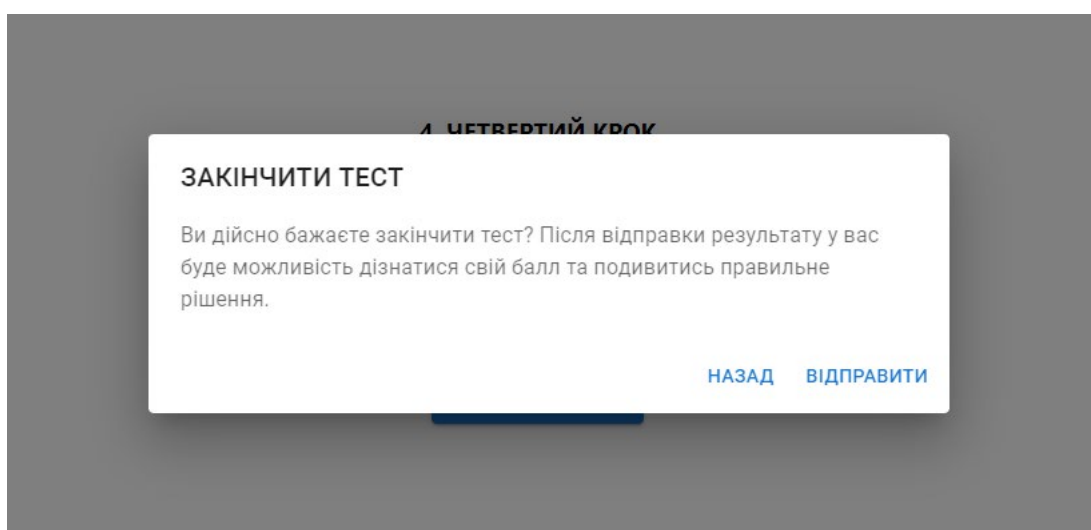


Рисунок 4.8 – Зовнішній вигляд модального вікна для підтвердження завершення тесту

Після завершення тесту у користувача є можливість переглянути результати тесту (рисунок 4.9).

КАЛЬКУЛЯТОР

ТЕСТ

**ВАШ БАЛ: 59**

На першому кроці ви не знайшли такі пари:  
[4 - 7], [6 - 7]

Пари, які ви знайшли неправильно:  
[7 - 7], [8 - 8]

За перший крок ви отримуєте - 21 балів із 30

---

На другому кроці ви не знайшли такі імпліканти:  
[x100]

Імпліканти, які ви знайшли неправильно:  
[xxxx]

За другий крок ви отримуєте - 16 балів із 25

---

На третьому кроці ви не знайшли такі імпліканти, які належать до ядра:  
[1x1x]

Імпліканти, які ви знайшли неправильно:  
[1111]

За третій крок ви отримуєте - 12 балів із 25

Рисунок 4.9 – Ілюстрація секції з результатами користувача

Приклад надісланого викладачеві електронного повідомлення, після завершення тесту користувачем наведений на рисунку 4.9.

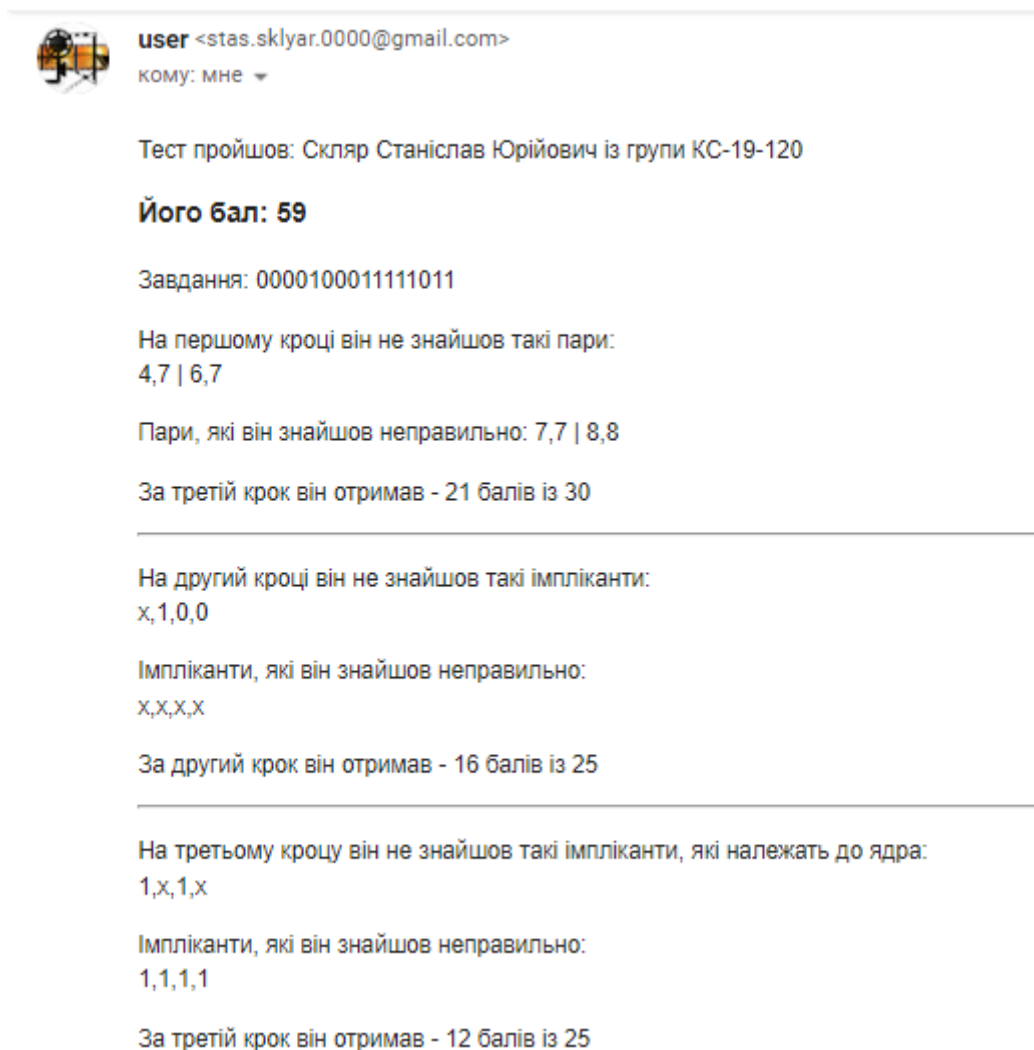


Рисунок 4.10 – Зовнішній вигляд електронного повідомлення надісланого викладачеві після завершення тесту

## 4.2 Результат роботи демонстраційного режиму НК

Графічна опис демонстраційного режиму навчального комплексу наведений в додатку Б.

Після переходу за посиланням користувач потрапляє на сторінку на якій в текстовому полі потрібно ввести задачу для демонстрації подальшої мінімізації (рисунок Б.1).

Після введення задачі потрібно натиснути кнопку «МІНІМІЗУВАТИ». Далі для демонстрації всіх етапів мінімізації, потрібно лише натискати кнопку «НАСТУПНИЙ КРОК».

## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи була досягнута головна мета, що була складовою аналізу головного завдання та формування постановки задачі.

Головним результатом роботи можна вважати web-застосунок, як програмна реалізація навчального комплексу «Мінімізація булевих функцій» та інструкція користувача, що допоможе достатньо легко інтегрувати web-застосунок до навчального процесу в межах дисципліни «Комп'ютерна логіка».

В даній кваліфікаційній роботі розроблено навчальний комплекс, що повинен допомогти студентам самостійно освоїти один з методів мінімізації, а саме: метод Квайна Мак-Класкі.

В процесі розробки навчального комплексу був виконаний ряд задач і отримані важливі результати, а саме:

- в результаті аналізу методу мінімізації булевих функцій визначені ключові моменти для навчально-демонстраційного та тестового режимів роботи навчального комплексу;
- Спираючись на постановку задачі було вирішені такі проблеми як:
  - 1) вибір програмної платформи;
  - 2) вибір середовища програмування;
  - 3) вибір мови програмування;
  - 4) аналіз та вибір сторонніх сервісів (бібліотек), які допоможуть реалізувати необхідний функціонал навчального комплексу;
- розроблені алгоритми для програмної реалізації НК;
- розроблена програмна реалізація;
- складена інструкція по використанню для двох режимів НК.

На основі програмної реалізації метода КМ створений веб додаток з двома режимами роботи:

- демонстраційний;
- тестовий.

В демонстраційному режимі реалізовано максимальну ілюстрацію та текстове супроводження всіх етапів мінімізації, з точки зору, навчальної користі пояснення матеріалу.

В тестовому режимі реалізовано відповідний інтерфейс, автоматичну систему оцінювання, можливість зберігання відповідей та підсумкової оцінки користувача.



## ПЕРЕЛІК ПОСИЛАНЬ

1. Лекція «Мінімізація булевих функцій» [Електронний ресурс]  
Режим доступу: <https://epa.kpi.ua/wp-content/Buryan/%D0%90%D0%A2%D0%9F%D0%A3-1.%20%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%203.pdf>
2. «Quine–McCluskey algorithm» [Електронний ресурс] Режим доступу: [https://en.wikipedia.org/wiki/Quine%E2%80%93McCluskey\\_algorithm](https://en.wikipedia.org/wiki/Quine%E2%80%93McCluskey_algorithm)
3. «Quine-McCluskey Tabular Method» [Електронний ресурс] Режим доступу:  
[https://www.tutorialspoint.com/digital\\_circuits/digital\\_circuits\\_quine\\_mccluskey\\_tabular\\_method.htm](https://www.tutorialspoint.com/digital_circuits/digital_circuits_quine_mccluskey_tabular_method.htm)
4. «Мобільний застосунок» [Електронний ресурс] Режим доступу:  
[https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B1%D1%96%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9\\_%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B1%D1%96%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA)
5. «Особенности тестирования десктопных додатків у порівнянні з Web та мобільними додатками» [Електронний ресурс] Режим доступу:  
<https://www.quality-assurance-group.com/osoblyvosti-testuvannya-desktopnyh-dodatkov-u-porivnyanni-z-web-ta-mobilnymy-dodatkamyy/>
6. «Mobile application development» [Електронний ресурс] Режим доступу:  
<https://www.techtarget.com/searchapparchitecture/definition/mobile-application-development><sup>7</sup> <https://www.centum-d.com/veb-dodatok-yogo-harakteristiki/>
7. «Веб-додаток і його характеристики» [Електронний ресурс] Режим доступу:  
<https://www.centum-d.com/veb-dodatok-yogo-harakteristiki/>
8. «Мова JavaScript та її можливості» [Електронний ресурс] Режим доступу:

<https://sites.google.com/site/webtehnologiietawebdizajn/mova-javascript-ta-ieie-mozlivosti>

9. «Что лучше всего изучать в 2020 году: Angular, React, или Vue.js»

[Электронный ресурс] Режим доступа:

<https://badcode.ru/chto-luchshie-vsiegho-izuchat-v-2020-ghodu-angular-react-ili-vue-js/>

10. «What is JavaScript?» [Электронный ресурс] Режим доступа:

[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)

11. Офіційний сайт бібліотеки Material UI [Електронний ресурс] Режим доступу:

<https://mui.com/>

12. Офіційний сайт бібліотеки Email JS [Електронний ресурс] Режим доступу:

<https://www.emailjs.com/>

13. «Git. Коротко о главном» [Электронный ресурс] Режим доступа:

<https://habr.com/ru/post/588801/>