

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

**на тему: «РОЗРОБКА АЛГОРИТМУ І ПРОГРАМИ
АВТОМАТИЗАЦІЇ ПРОЦЕСУ ПРОГНОЗУВАННЯ
ВЛАСТИВОСТІ ГРАФІТИЗОВАНИХ СТАЛЕЙ ДЛЯ
КОНКРЕТНИХ УМОВ ЕКСПЛУАТАЦІЇ»**

Виконав: студент 2 курсу, групи 8.1211-з

спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)

освітньої програми інженерія програмного забезпечення
(назва освітньої програми)

І.О. Маляревич

(ініціали та прізвище)

Керівник декан математичного факультету,
професор, д.т.н. Гоменюк С.І.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри фундаментальної та прикладної
математики, професор, д.т.н. Гребенюк С.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти магістр

Спеціальність 121 інженерія програмного забезпечення

Освітня програма інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент

Лісняк А.О.

(підпис)

« _____ » _____ 2022 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Маляревичу Івану Олександровичу

(прізвище, ім'я та по-батькові)

1. Тема роботи (проєкту) Розробка алгоритму і програми автоматизації процесу прогнозування властивості графітизованих сталей для конкретних умов експлуатації

керівник роботи (проєкту) Гоменюк Сергій Іванович, д.т.н., професор
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затвержені наказом ЗНУ від « 10 » травня 2022 року № 514-с

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік питань до розробки.
3. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі, аналіз предметної області.
2. Приклади використання інформаційної системи.
3. Переваги користування інформаційною системою.

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	12.05.2022	
2.	Збір вихідних даних.	28.05.2022	
3.	Обробка методичних та теоретичних джерел.	06.09.2022	
4.	Розробка першого та другого розділів.	17.10.2022	
5.	Розробка третього та четвертого розділів.	21.11.2022	
6.	Оформлення та нормоконтроль кваліфікаційної роботи магістра.	02.12.2022	
7.	Захист кваліфікаційної роботи.	16.12.2022	

Студент

_____ (підпис)

І.О. Маляревич

_____ (ініціали та прізвище)

Керівник роботи

_____ (підпис)

С.І. Гоменюк

_____ (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер

_____ (підпис)

А.В. Столярова

_____ (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка алгоритму і програми автоматизації процесу прогнозування властивості графітізованих сталей для конкретних умов експлуатації»: 42 с., 15 рис., 4 табл., 10 джерел, 1 додаток.

ДОДАТОК, ПРОГНОЗУВАННЯ, РОЗРОБКА, СТАЛЬ ГРАФІТИЗОВАНА, ANGULAR, NODE.JS, WEB.

Об'єкт дослідження – процес розробки алгоритму і програми для прогнозування властивостей графітізованої сталі з різним вмістом вуглецю.

Мета роботи: розробка алгоритму і програми для прогнозування властивостей графітізованих сталей, шляхом побудови математичної моделі на основі отриманих зразків з різним вмістом вуглецю в широкому діапазоні та їх властивостей.

Метод дослідження – аналітичний.

У кваліфікаційній роботі показано створену математичну модель, для прогнозування властивостей графітізованих сталей, на підґрунті тих даних, що були отримані експериментальним шляхом, зокрема зносостійкість в умовах сухого тертя та тертя з змащування металу по металу.

SUMMARY

Master's qualifying paper «Development of Algorithm and Program for Automation of the Process of Predict the Properties of Graphitized Steels for Specific Operating Conditions»: 42 pages, 15 figures, 4 tables, 10 references, 1 supplement.

ANGULAR, APPLICATION, DEVELOPMENT, FORECASTING, GRAPHITED STEEL, NODE.JS, WEB.

Object of study – the process of developing an algorithm and program for predicting the properties of graphitized steel with different carbon content.

The goal of the work: development of an algorithm and program for predicting the properties of graphitized steels by building a mathematical model based on the obtained samples with different carbon content in a wide range and their properties.

Research method – analytical.

The qualification paper shows the created mathematical model for predicting the properties of graphitized steels, based on the data obtained experimentally, in particular, wear resistance under conditions of dry friction and friction from metal-to-metal lubrication.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат.....	4
Summary.....	5
Вступ.....	8
1 Прогнозування, моделі, методи та алгоритми	9
1.1 Загальні відомості	9
1.2 Методи прогнозування.....	9
1.3 Використання в бізнесі.....	10
1.4 Діджиталізація в сучасній металургії	11
2 Аналіз та оцінка існуючих систем прогнозування	12
2.1 Аналіз існуючих систем.....	12
2.2 Переваги існуючих систем.....	14
2.3 Недоліки існуючих систем.....	14
2.4 Аналіз на ринку	15
3 Інструменти розробки, фреймворк, мова розробки та бібліотеки	16
3.1 Вибір технології	16
3.2 Робоче середовище	16
3.3 Фреймворк та інструмент розробки.....	18
3.4 Мова розробки.....	18
3.5 Бібліотека Google Charts	20
3.6 Бібліотека Lodash	20
4 Реалізація застосунку для прогнозування властивостей графітизованих сталей	22
4.1 Механічні властивості графітизованих сталей	22
4.2 Дослідження зносостійкості ГС в умовах сухого тертя та тертя зі змащуванням метал по металу	22

4.3	Визначення математичної моделі для прогнозування властивостей графітизованих сталей.....	25
4.4	Побудування алгоритму для прогнозування властивостей графітизованих сталей	26
4.5	Реалізація додатку.....	27
	Висновки	31
	Перелік посилань.....	32
	Додаток А Лістинг програмного коду.....	33

ВСТУП

Зниження матеріаломісткості при виробництві деталей машин – це одна з головних тенденцій розвитку промисловості, але при одночасному підвищенні надійності, довговічності цих деталей. Ця мета досягається, в основному, в результаті коректного вибору матеріалу.

Виробництво графітізованих сталей (ГС) в Україні практично не налагоджено, відсутні технічні вимоги й державні стандарти, що регламентували б їх склади і вимоги до якості.

Для коректного вибору матеріалу потрібно знати які він має властивості і як він поводить себе в різних умовах експлуатації.

Раніше для отримання і зберігання інформації використовували таблиці та довідники. Але у сучасних умовах, ми можемо використовувати обчислювані потужності комп'ютерів. Сучасні прилади з легкістю можуть зберігати надзвичайну велику кількість інформації, зокрема такі як таблиці та довідники, та дають можливість отримувати її дуже швидко.

Якщо працювати з мало дослідженими матеріалами або експериментальними матеріалами тоді потрібно створити документацію та провести багату кількість експериментів, а це вже значно здорожчує виробництво. Виходячи з цього, дешевше та швидше навчити машину прогнозувати властивості матеріалу базуючись на результатах декількох експериментів, а для цього потрібно розробити алгоритм. Результат розробленого алгоритму повинен збігтися або бути в допустимих похибках з результатами практичних експериментів.

1 ПРОГНОЗУВАННЯ, МОДЕЛІ, МЕТОДИ ТА АЛГОРИТМИ

1.1 Загальні відомості

Прогнозування – це оцінка величини невизначених майбутніх подій і надання різних результатів з різними припущеннями. Найпопулярніші методи прогнозування включають якісне прогнозування і кількісне прогнозування. Не всі методи обов’язково слугуватимуть цілям прогнозування, особи, які приймають рішення, повинні розуміти, який тип найкраще підходить для бізнесу.

Прогнозування являє собою складний процес, під час якого необхідно вирішувати велику кількість різних питань. Для його виробництва слід застосовувати в поєднанні різні методи прогнозування, яких на сьогоднішній день існує безліч, але на практиці використовуються не більше двадцяти.

1.2 Методи прогнозування

Методи прогнозування можна в цілому класифікувати на якісні та кількісні.

Якісні методи – ці методи базуються на емоціях, інтуїції, судженнях, особистому досвіді та думках. Це означає, що в якісних методах прогнозування немає математики. Частиною цього типу є:

- метод Delphi;
- market survey;
- executive opinion;
- salesforce composite.

Кількісні методи – ці методи повністю залежать від математичних або кількісних моделей. Результат цього методу повністю залежить від математичних розрахунків. До цього типу належать такі моделі як:

- часові ряди;
- асоціативні моделі.

1.3 Використання в бізнесі

Прогнозування дозволяє компанії вжити необхідних заходів для досягнення конкретної мети, надаючи важливу інформацію щодо майбутніх подій, їх виникнення та масштабів. Прогнозування може бути як якісним, так і кількісним, залежно від зібраної інформації та її характеру, як правило, суб'єктивного або об'єктивного характеру, і, як наслідок, ґрунтується на математичних розрахунках або взагалі не ґрунтується на математичних розрахунках.

Керівництво вирішує найкращий метод прогнозування, який буде використовуватися відповідно до бізнесу. Він базується на внутрішніх і зовнішніх факторах і на тому, чи є зовнішні фактори контрольованими чи неконтрольованими. Неконтрольованими факторами можуть бути державна політика, стратегія конкурентів, стихійні лиха тощо. Кількісне прогнозування використовує математичні моделі для отримання результатів прогнозування, і він також спирається на історичні дані, щоб підтвердити висновки. Якісне прогнозування використовує емоції, інтуїцію, минулий досвід і цінності. Це важлива процедура в бізнесі, яка покращує бізнес-операції та забезпечує безперебійне виконання функцій у бізнес-середовищі, що постійно змінюється [1].

1.4 Діджиталізація в сучасній металургії

Сьогодні визначальним чинником у підтримці конкурентоспроможності металургії на світовому ринку є технологічні інновації, цифрові технології в металовиробництві.

В сучасній металургії з'явився ще один напрямок – діджиталізація. Даний напрямок значно підвищить безпеку на робочому місці завдяки повній автоматизації та використанню у небезпечних робочих зонах роботів. Технічне обслуговування полегшить системи моніторингу стану (CMS) та доповнена реальність (AR). Завдяки допомозі штучного інтелекту більшість процесів будуть оптимізовані, а дефекти кінцевих продуктів стануть.

При розвитку новітніх технологій, традиційні не можна списувати з рахунків, так як вони вдосконалюються та модернізуються з метою підвищення якості продукції та забезпеченні конкурентоспроможності на перенасиченому ринку.

Як відзначають автори [2], Web-сервіси все активніше проникають в область, вже зайняту об'єктно і компонентними технологіями, представляючи собою альтернативу традиційним підходам до створення корпоративних додатків. В останні роки інвестицій в технології, пов'язані з Web-сервісами, збільшилися. Так як все більше технологій проникають в наше життя, розробка додатків на багато платформ (комп'ютери, смартфони, смартТВ і інше) за допомогою однієї технології куди перспективнішою, з боку розробки та економічної, ніж вибір технології під кожну платформу.

Одна з головних переваг програмного прогнозування це те, що не потрібно робити експериментальні зразки для подальшого руйнування. Собівартість такого способу встановлення властивостей матеріалу, так само і економічна ефективність, значно нижче [3].

2 АНАЛІЗ ТА ОЦІНКА ІСНУЮЧИХ СИСТЕМ ПРОГНОЗУВАННЯ

2.1 Аналіз існуючих систем

Цей розділ містить огляд і оцінку деяких безкоштовних або умовно-безкоштовних систем, які існують на ринку. Розглянуто підходи та зроблено оцінку, також розглянуто систему з точки зору користувача.

На рисунках 2.1 – 2.3 представлені інтерфейси існуючих систем, таких як:

- JmatPro;
- SOLIDWORKS Materials;
- Ansys GRANTA MI.

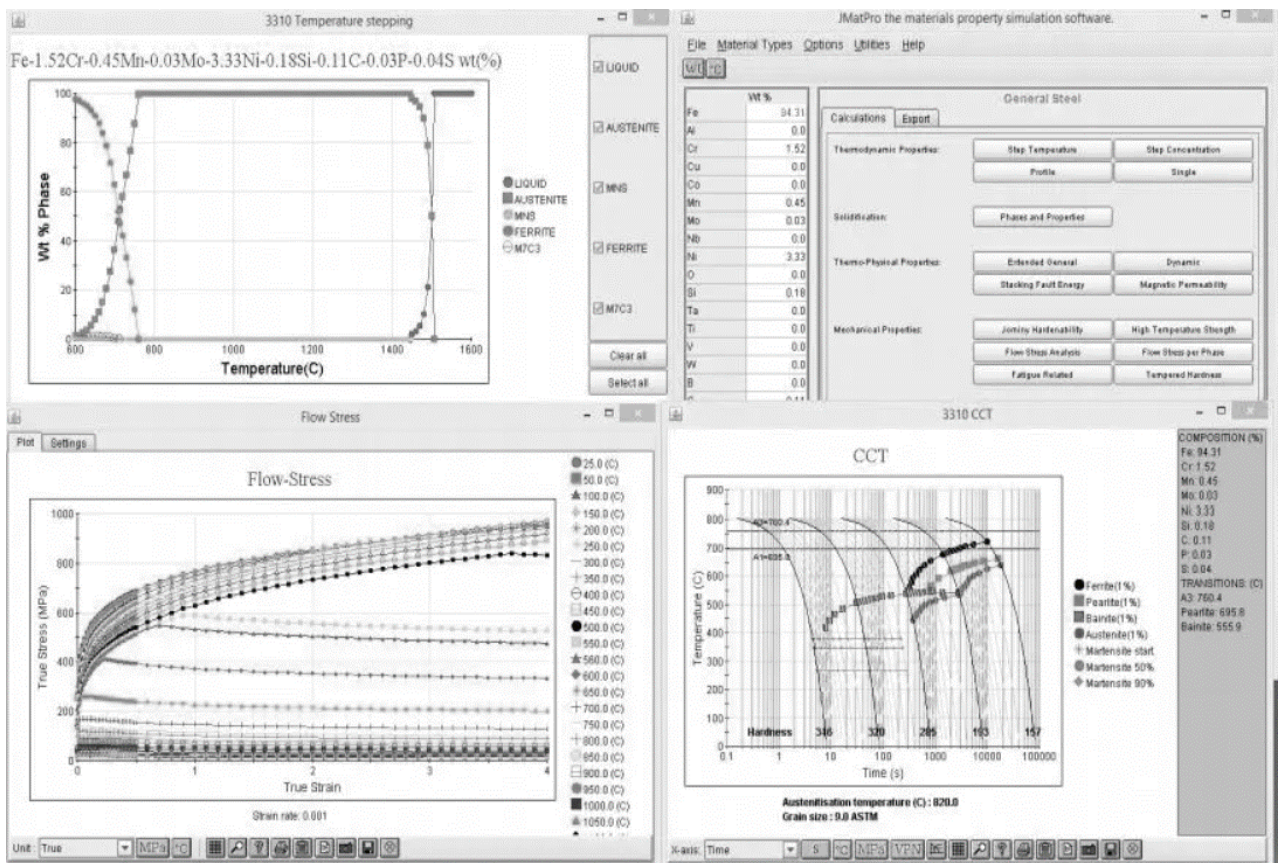


Рисунок 2.1 – Інтерфейс JmatPro [4]

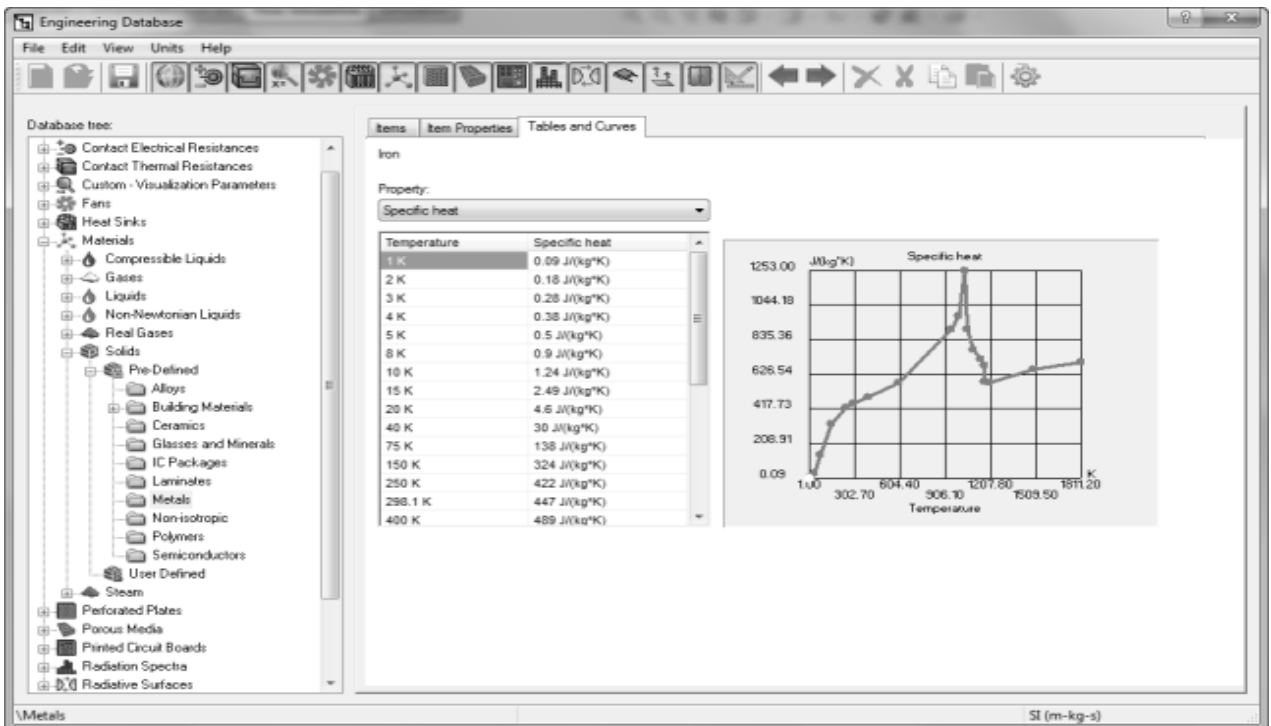


Рисунок 2.2 – Інтерфейс SOLIDWORKS Materials [5]

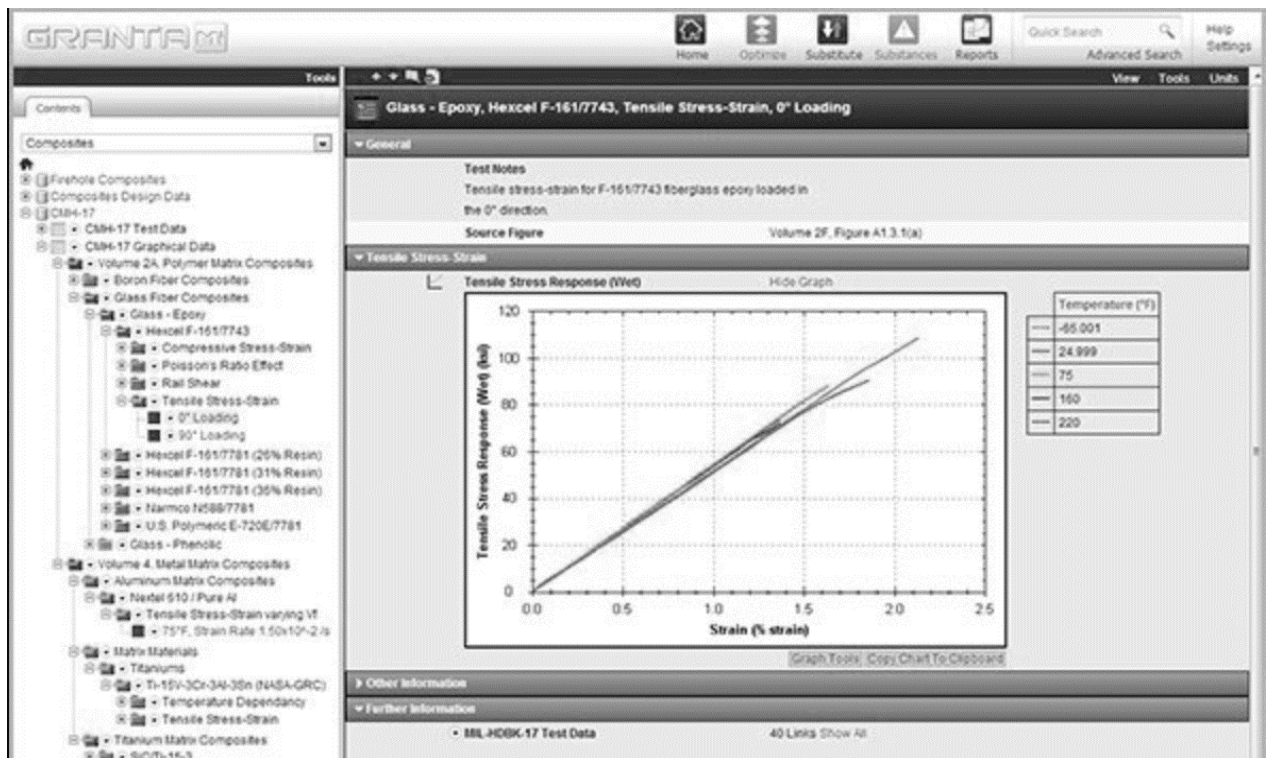


Рисунок 2.3 – Інтерфейс Ansys GRANTA MI [6]

2.2 Переваги існуючих систем

Якщо проаналізувати наявні продукти на ринку, ми можемо побачити, що найбільш інформативними та й дружніми для користувачів є системи які базуються на візуальному відображенні за допомогою графіків та діаграм. При цьому підходу користувач, має змогу оперувати вже не тільки цифрами, але й їх візуальних відображенням, щоб краще розуміти різницю між певними показниками.

2.3 Недоліки існуючих систем

Одними з найголовніших недоліків існуючих на ринку систем для роботи з прогнозування властивостей різноманітних матеріалів, можна виділити такі як:

- складний шлях для початку роботи з калькулятором;
- не дружелюбний інтерфейс;
- відсутність прогнозування для ГС;
- немає модулю, що запускається окремо від застосунку.

Складний шлях для початку роботи з калькулятором – такий недолік не дозволяє користувачу навіть побачити цей функціонал, що зводить нанівець всі потуги розробників, які намагалися імплементувати цей функціонал.

Не дружелюбний інтерфейс – користувач не може зорієнтуватися як саме використовувати той чи інший компонент, та більшість функціоналу є не інтуїтивними, а більшість патернів, які базуються на UI&UX, повсякчас порушуються, що робить додаток дуже складним у використанні пересічними користувачами.

Відсутність прогнозування для ГС – в усіх наведених вище додатках, які існують на ринку, нажаль, відсутній головний функціонал, який потрібен при роботі з ГС, що робить його менш корисним для роботи з даним матеріалом в єдиній системі.

2.4 Аналіз на ринку

Проаналізувавши всі недоліки та переваги існуючих систем для прогнозування властивостей матеріалів, можемо зробити такі висновки, що наш застосунок повинен мати наступні головні та вторинні характеристики (див. табл. 2.1).

Таблиця 2.1 – Необхідні характеристики для застосунку

Характеристика	Пріоритет
Дружелюбність додатку	8
Візуальне відображення даних за допомогою графіків	10
Автономність застосунків	4
Складний шлях для початку роботи з калькулятором	8
Наявність прогнозування для ГС	10

3 ІНСТРУМЕНТИ РОЗРОБКИ, ФРЕЙМВОРК, МОВА РОЗРОБКИ ТА БІБЛІОТЕКИ

3.1 Вибір технології

Обрали веб технології для розробки додатку щоб побудувати математичну модель та обробляти розрахунки використовуючи потужності веб-браузерів. Веб-сайти – основа всесвітньої павутини що дає можливість зберігати математичну модель не на комп'ютері, а в мережі інтернет (рис. 3.1).

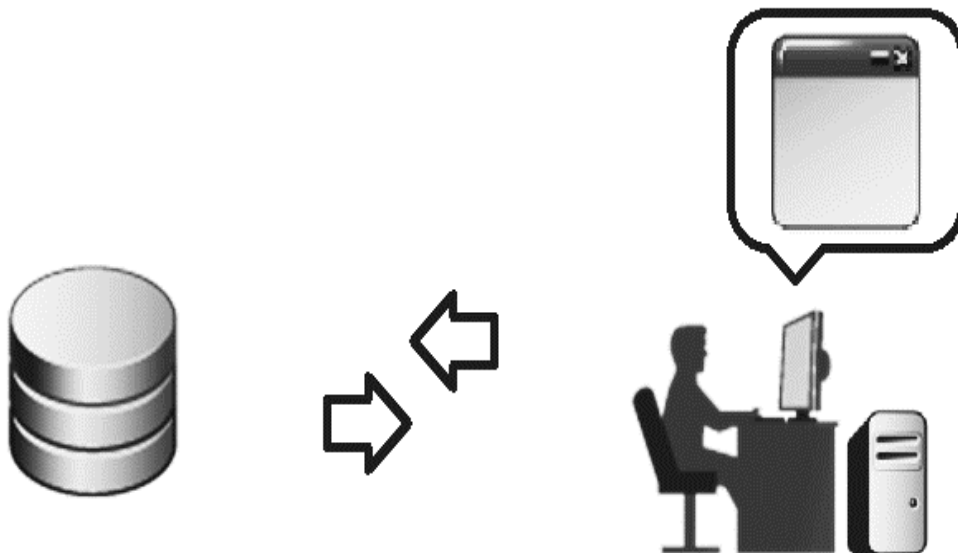


Рисунок 3.1 – Схема роботи Сервера та клієнта-браузера

3.2 Робоче середовище

Для того щоб зробити розробку приємнішою, автоматизувати рутинну роботу та з легкістю вирішувати складні завдання, було вирішено використовувати середовище розробки WebStorm. WebStorm – це інтегроване

середовище розробки для кодування на JavaScript і пов'язаних із ним технологіях, включаючи TypeScript, React, Vue, Angular, Node.js, HTML і таблиці стилів. Так само, як IntelliJ IDEA та інші IDE JetBrains, WebStorm.

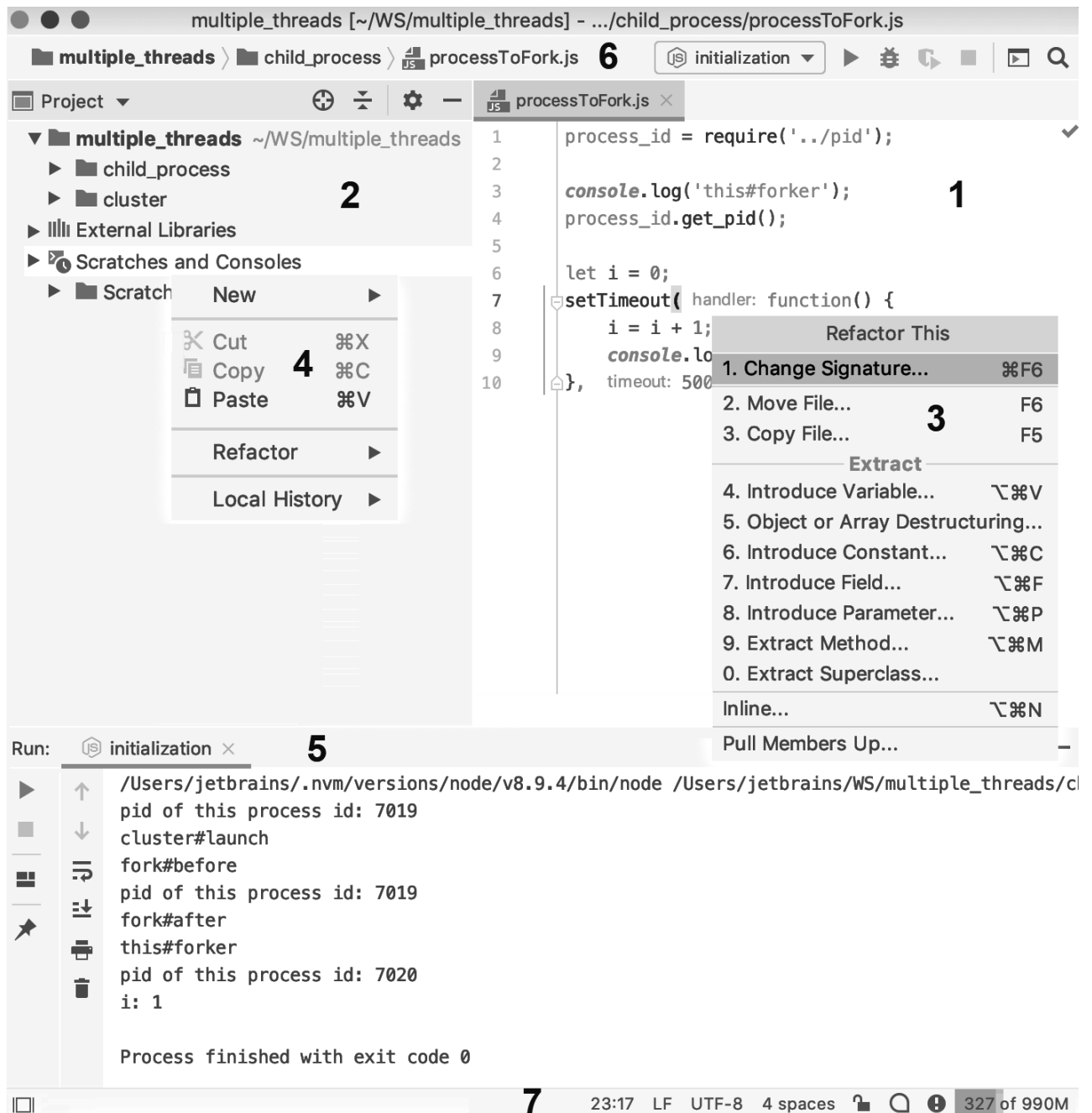


Рисунок 3.2 – Інтерфейс користувача WebStorm

При відкритті проекту WebStorm, інтерфейс користувача (рис. 3.2) за замовчуванням складаються із семи елементів:

- 1) редактор;

- 2) вікно інструментів проекту;
- 3) спливаюче меню;
- 4) контекстне меню;
- 5) вікно інструментів запуску;
- 6) панель навігації;
- 7) рядок стану.

3.3 Фреймворк та інструмент розробки

Для додатку обрали Angular [7] для клієнтської сторони та Node.js [8] для серверної.

Для розробки клієнтської сторони веб-додатків обрали фреймворк Angular з відкритим вихідним кодом, заснованого на TypeScript та технічною підтримкою корпорації Google і співтовариством приватних осіб і компаній.

Для розробки серверної сторони веб-додатків обрали кросплатформове середовище виконання JavaScript з відкритим вихідним кодом, яка може виконувати код JavaScript поза веб-браузера та дозволяє розробникам використовувати JavaScript для написання інструментів командного рядка і виконання сценаріїв на стороні сервера, тобто запуску сценаріїв на стороні сервера для створення динамічного веб-контенту перед відправкою сторінки в веб-браузер користувача.

3.4 Мова розробки

TypeScript є мовою програмування з відкритим кодом. Він містить набір функцій об'єктно-орієнтованого програмування (ООП), які допомагають підтримувати якість коду. Ви можете думати про це як про версію JavaScript ES6 із деякими додатковими функціями, як-от суворе прив'язування типів.

Код TypeScript компілюється до JavaScript. Це відбувається тому, що браузер не може зрозуміти TypeScript, вони розуміють лише JavaScript. Через це TypeScript спочатку потрібно скомпілювати в JavaScript, потім цей скомпільований JavaScript виконується в браузері.

Компіляція є однією з головних переваг TypeScript над JavaScript. Так як JavaScript являє собою інтерпретовану мову, це означає, щоб перевірити, чи все працює належним чином, потрібно запуснути його в браузері. Якщо є якісь помилки, їх потрібно знайти та налагодити під час написання коду, а отже, це забирає більше часу та зусиль. В свою чергу, TypeScript забезпечує зручність перевірки помилок під час компіляції. Якщо виникають якісь синтаксичні помилки, TypeScript скомпілює ці помилки та повідомить про них перед виконанням сценарію, позбавляючи вас від роботи з помилками під час виконання, які, як правило, важче визначити та налагодити.

Процес рефакторингу надійний і легший у TypeScript, ніж у порівнянні з JavaScript. Так як TypeScript має «типи», які забезпечують гнучкість під час рефакторингу коду. Він виявляє помилки, пов'язані з типом, під час компіляції, а не чекає виняткової ситуації під час виконання.

У TypeScript є такі базові типи [9]:

- Boolean – логічне значення true чи false;
- Number – числове значення;
- String – строка;
- Array – масиви;
- Enum – перерахування;
- Any – довільний тип;
- Symbol;
- null і undefined – відповідають значенням null і undefined у JavaScript;
- Never – також представляє відсутність значення і використовується як тип функцій, що повертаються, які генерують або повертають помилку.

3.5 Бібліотека Google Charts

Для малювання графіку та роботи з ним було обрано обгортка для бібліотеки Google Charts, написана мовою Angular, під назвою `angular-google-charts`.

Google Charts надає ідеальний спосіб візуалізації даних на вашому веб-сайті. Від простих лінійних діаграм до складних ієрархічних деревних карт, галерея діаграм надає велику кількість готових до використання типів діаграм.

Найпоширенішим способом використання Google Charts є простий JavaScript, який ви вбудовуєте у свою веб-сторінку. Ви завантажуєте деякі бібліотеки Google Chart, перераховуєте дані для діаграми, вибираєте параметри для налаштування діаграми та, нарешті, створюєте об'єкт діаграми з ідентифікатором, який ви вибираєте. Потім пізніше на веб-сторінці ви створюєте `<div>` із цим ідентифікатором для відображення Google Chart [10].

3.6 Бібліотека Lodash

Для забезпечення більшої продуктивності при розробці було використано Lodash – сучасну бібліотеку утиліт JavaScript. Вона полегшує роботу з JavaScript та всіма похідними від нього мовами. Він допомагає працювати з масивами, рядками, об'єктами, числами тощо.

Lodash надає різноманітні вбудовані функції та використовує підхід до функціонального програмування, який легше зрозуміти з кодуванням у JavaScript, оскільки замість написання повторюваних функцій завдання можна виконувати за допомогою однієї рядок коду. Це також полегшує роботу з об'єктами в JavaScript, якщо з ними потрібно багато маніпулювати.

Модульні методи Lodash чудово підходять для:

- ітерації масивів, об'єктів і рядків;
- маніпулювання та тестування значень;

- створення складених функцій.

Недоліки Lodash:

- найбільший недолік Lodash – швидкість: для виконання методів Lodash потрібно більше часу, ніж для звичайних функцій JavaScript;
- у JavaScript ми можемо об'єднати багато функцій, як забажаємо, але в Lodash ми не можемо об'єднати функції, ми можемо лише обернути їх.

Переваги Lodash:

- він надає нам різноманітні вбудовані функції для колекцій, масивів, для маніпулювання об'єктами та іншими службовими методами, які ми можемо використовувати безпосередньо замість того, щоб писати їх з нуля;
- це робить наш код більш коротким і зрозумілим, що полегшує його розуміння та зміну пізніше;
- ми повинні пам'ятати про функції Lodash, і це значно спрощує завдання кодування;
- початківцям у техніці легше почати й навчатися;
- замість того, щоб писати повторювані функції, завдання можна виконувати за допомогою одного рядка коду.

4 РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ДЛЯ ПРОГНОЗУВАННЯ ВЛАСТИВОСТЕЙ ГРАФІТИЗОВАНИХ СТАЛЕЙ

4.1 Механічні властивості графітизованих сталей

Механічні властивості ГС після графітизованого відпалу и гартування з низьким відпуском представлені в таблиці 4.1.

Таблиця 4.1 – Механічні властивості ГС після графітизованого відпалу и гартування з низьким відпуском

Номер випробування	Вміст С, %	Кількість графіту в структурі V_{β} , об., %	σ_B , МПа	δ , %	Твердість, НВ
1	0,48	4,97	1350	8 %	450
2	0,72	7,23	1900	6 %	480
3	1,25	8,59	1730	4 %	430
4	1,7	14,02	1240	2 %	410
5	1,95	16,29	1100	1 %	390

4.2 Дослідження зносостійкості ГС в умовах сухого тертя та тертя зі змащуванням метал по металу

В результаті експериментів зносостійкості ГС в умовах сухого тертя було отримано V_m – коефіцієнт втрати матеріалу (див. табл. 4.2), згідно формули:

$$V_m = m / t, \quad (4.1)$$

де V_m – швидкість втрати маси за одиницю часу;

m – маса втраченої частки зразка;

t – тривалість випробувань.

Таблиця 4.2 – Результати випробування на зносостійкість зразків ГС в умовах сухого тертя метал по металу

Номер випробувань	Вміст С, %	V_m	t тривалість випробувань
1	0,48	1,61	90 хвилин
2	0,72	1,32	
3	1,25	0,91	
4	1,7	0,56	
5	1,95	0,34	

В результаті експериментів зносостійкості ГС в умовах тертя з змащування метал по металу було отримано V_m – коефіцієнт втрати матеріалу (табл. 4.3).

Таблиця 4.3 – Результати випробування на зносостійкість зразків ГС в умовах тертя з змащування метал по металу

Номер випробувань	Вміст С %	V_m	t тривалість випробувань
1	0,48	0,86	90 хвилин
2	0,72	0,57	
3	1,25	0,22	

Продовження табл. 4.3

Номер випробувань	Вміст С %	V_m	t тривалість випробувань
4	1,7	0,11	90 хвилин
5	1,95	0,07	

Згідно отриманих даних, відображених в таблиці 4.2, побудуємо графік впливу кількості вуглецю у складі дослідних сплавів (С %) на втрату маси випробувальних зразків (V_m) в умовах сухого тертя метал по металу (рис. 4.1).

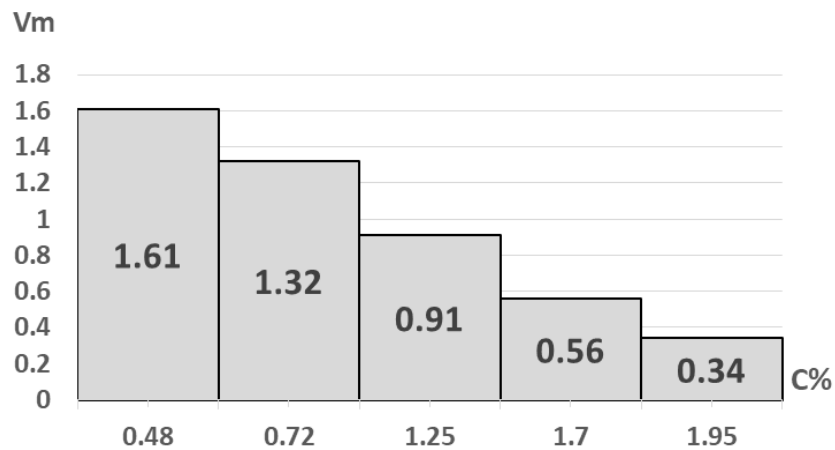


Рисунок 4.1 – Графік впливу кількості вуглецю у складі дослідних сплавів (С %) на втрату маси випробувальних зразків (V_m) в умовах сухого тертя метал по металу

Згідно отриманих даних, внесених в таблиці 4.3, побудуємо графік впливу кількості вуглецю у складі дослідних сплавів (С %) на втрату маси випробувальних зразків (V_m) в умовах тертя з змащування метал по металу (див. рис. 4.2).

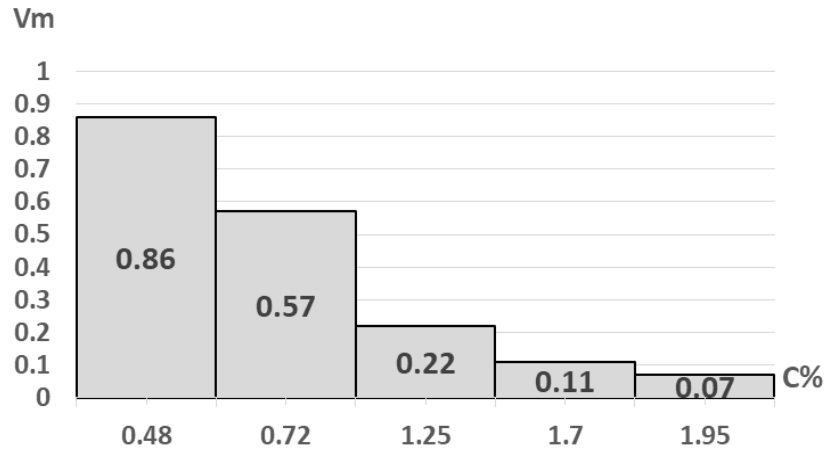


Рисунок 4.2 – Графік впливу кількості вуглецю у складі дослідних сплавів (С %) на втрату маси випробувальних зразків (V_m) в умовах тертя з змащування метал по металу

4.3 Визначення математичної моделі для прогнозування властивостей графітованих сталей

Побудову рівняння виду $V_m = f(C\%)$ проводили на базі чотирьох експериментальних точок для дослідів № 1, № 2, № 3, № 4 (див. табл. 4.2).

Після постановки мети і завдання додатку, були пройдені етапи створення та опрацювання технічного завдання (ТЗ) на розробку додатку (рис. 4.3).

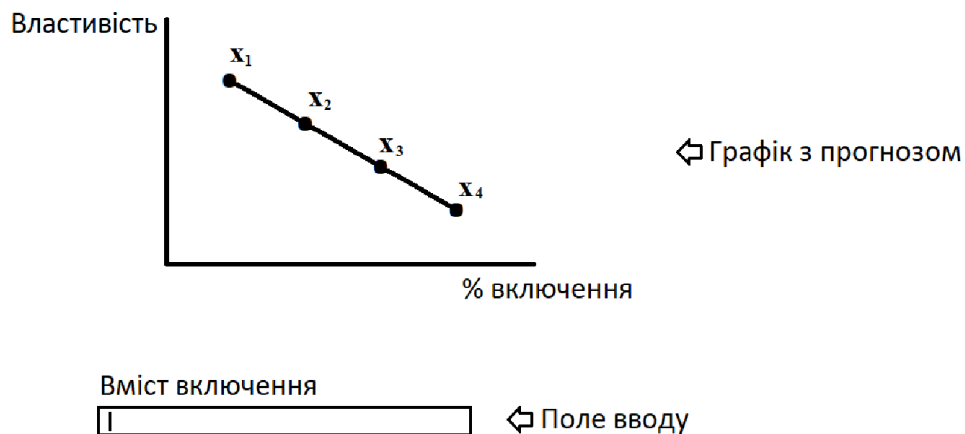


Рисунок 4.3 – Схематичне зображення додатка

Використовуючи обрані інструменти розробки Web-сервісів перейшли до розробки додатку.

4.4 Побудування алгоритму для прогнозування властивостей графітизованих сталей

Для прогнозування було обрано математичний метод інтерполяція, а саме Інтерполяційний многочлен Лагранжа – многочлен мінімального степеня, що приймає дані значення у даному наборі точок, тобто вирішує завданням інтерполяції. Його реалізація в коді зображено на рисунку 4.4.

```
lanPol(x: number, x_values: number[], y_values: number[], size: number) {  
  let lagrange_pol = 0;  
  let basics_pol;  
  for (let i = 0; i < size; i++) {  
    basics_pol = 1;  
    for (let j = 0; j < size; j++) {  
      if (j == i) continue;  
      basics_pol *= (x - x_values[j]) / (x_values[i] - x_values[j]);  
    }  
    lagrange_pol += basics_pol * y_values[i];  
  }  
  return lagrange_pol;  
}
```

Рисунок 4.4 – Інтерполяційний многочлен Лагранжа на мові програмування JavaScript

Функціонал та залежності веб-додатку що розробляється, зображений на діаграмі варіантів використання або діаграма Use Case (див. рис. 4.5).

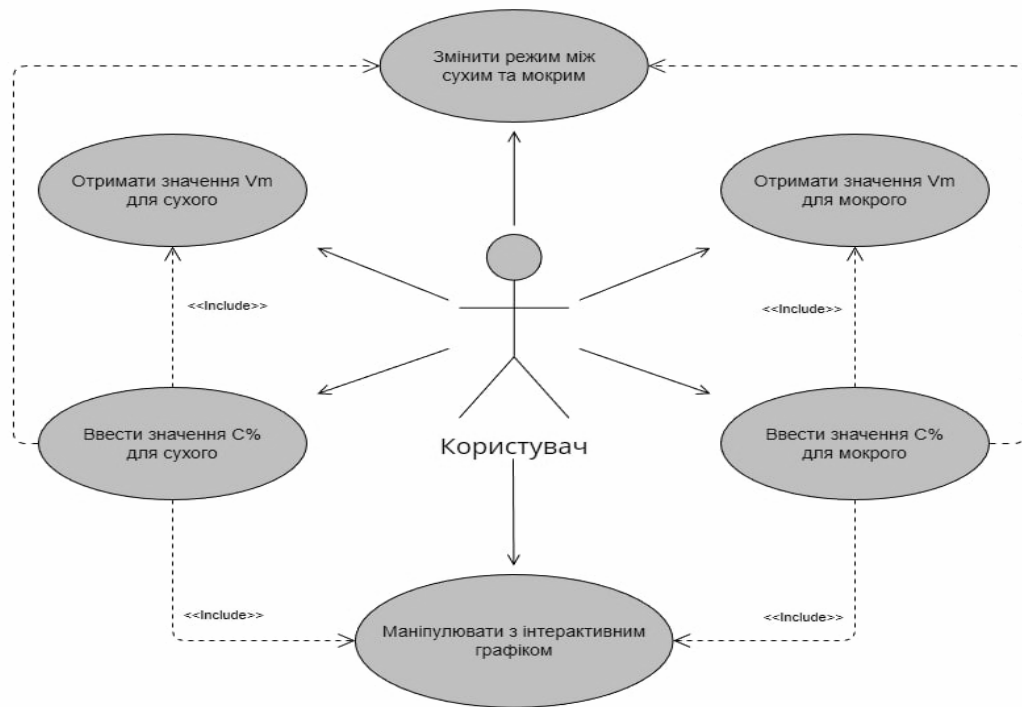


Рисунок 4.5 – Діаграма Use Case

4.5 Реалізація додатку

Створення макета дизайну додатку (див. рис. 4.6).

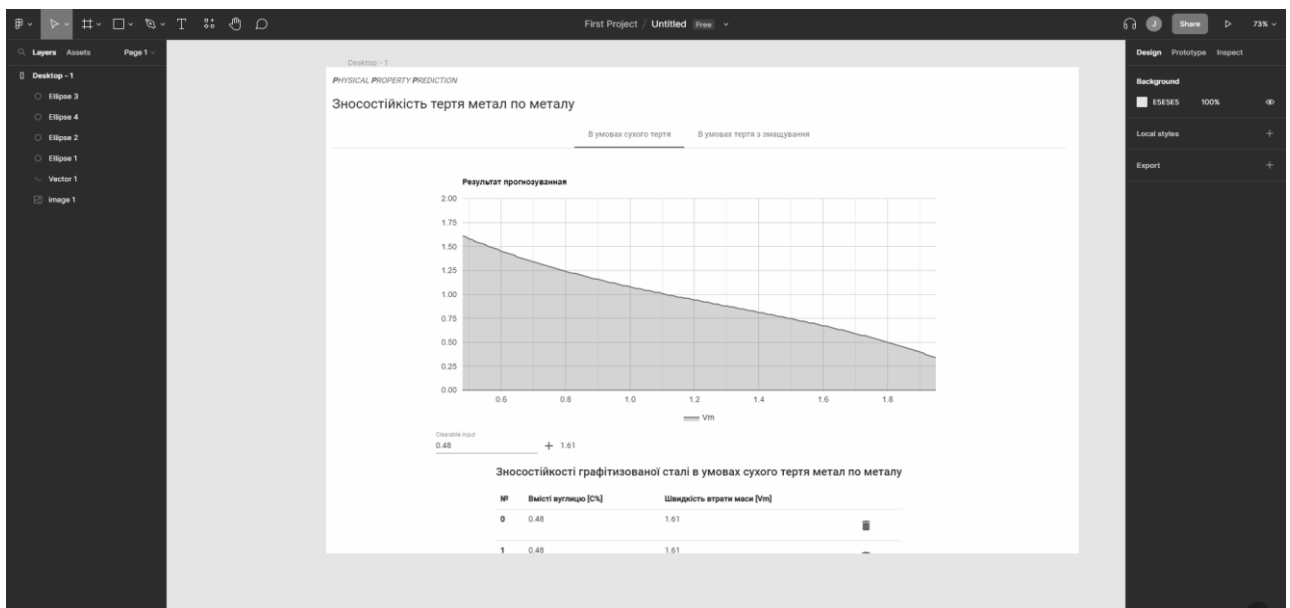


Рисунок 4.6 – Макет дизайн додатку

Верстку та програмування представлено на рисунках 4.7 та 4.8.

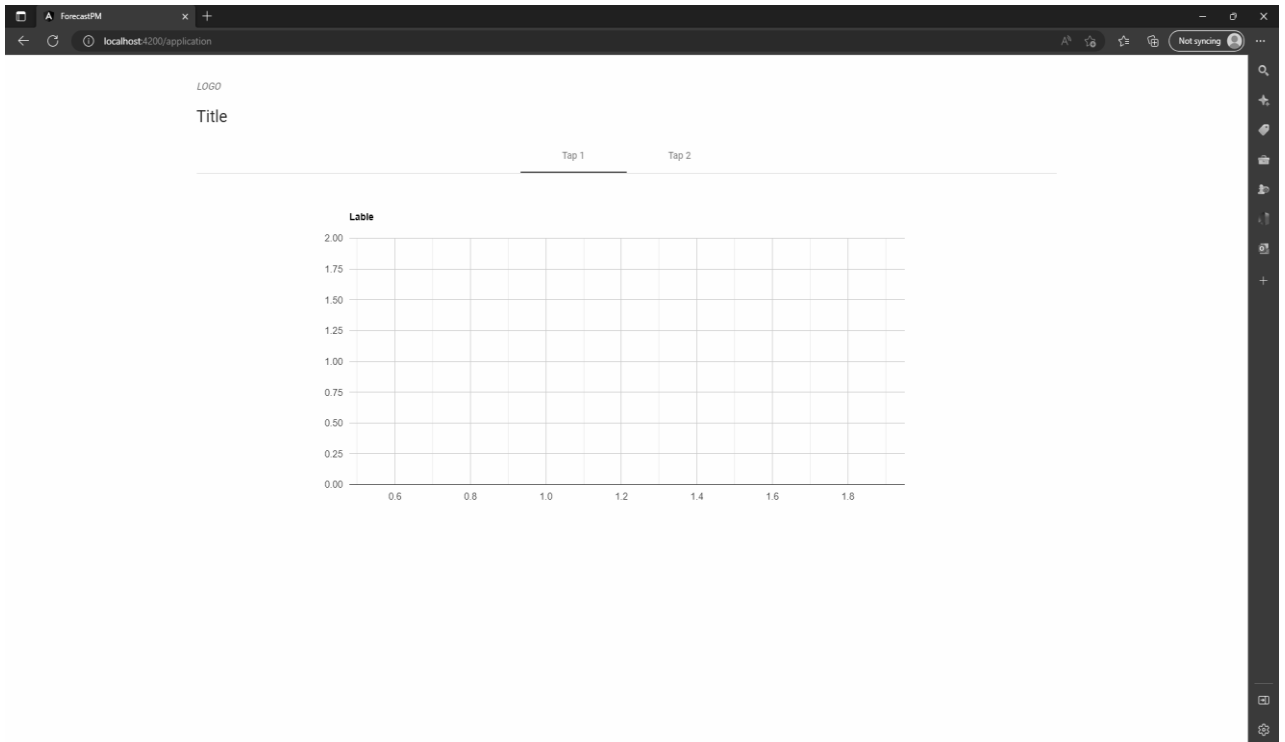
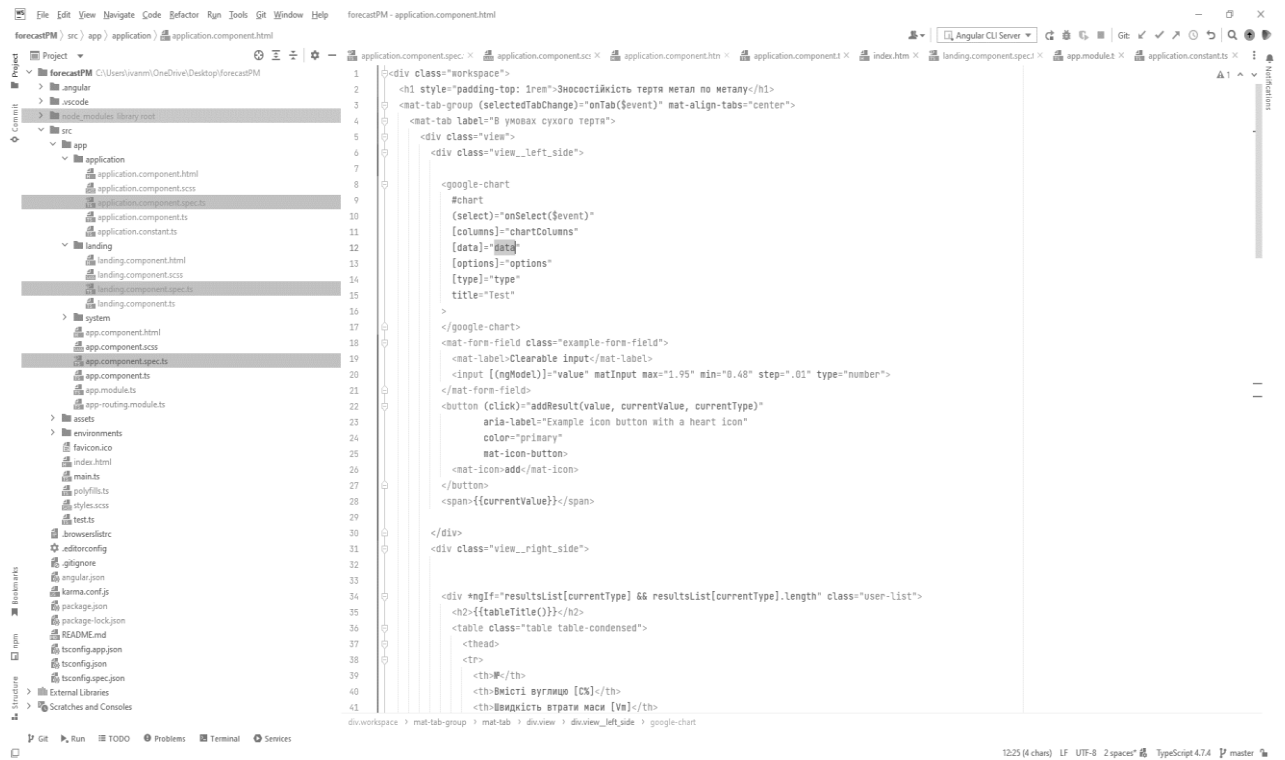


Рисунок 4.7 – Браузер на етапі верстки та програмування



Наповнення контентом та підключення бізнес логіки (рис. 4.9).



Рисунок 4.9 – Додаток на етапі наповнення контентом та підключення бізнес логіки

Графічне відображення рівняння наведено на рис 4.10. На цьому рисунку також наведено експериментальні точки опитів №1-5. З нього можна бачити, що перетин кривої розрахований рівнянням з вертикальним значенням 1,95 % майже не відрізняється за координатами (V_m ; C %) від точки отриманої експериментальним шляхом, а похибка знаходиться в межах 5 %. Це дає можливість зробити висновок, що отримана математична модель та її крива є здатною до прогнозування швидкості втрати маси V_m для ГС з вмістом C від 0,5 до 2,0 % після гартування та низького відпуску в умовах сухого тертя та тертя з змащування метал по металу.

Прогнозування за допомогою розробленого додатку (рис. 4.10).

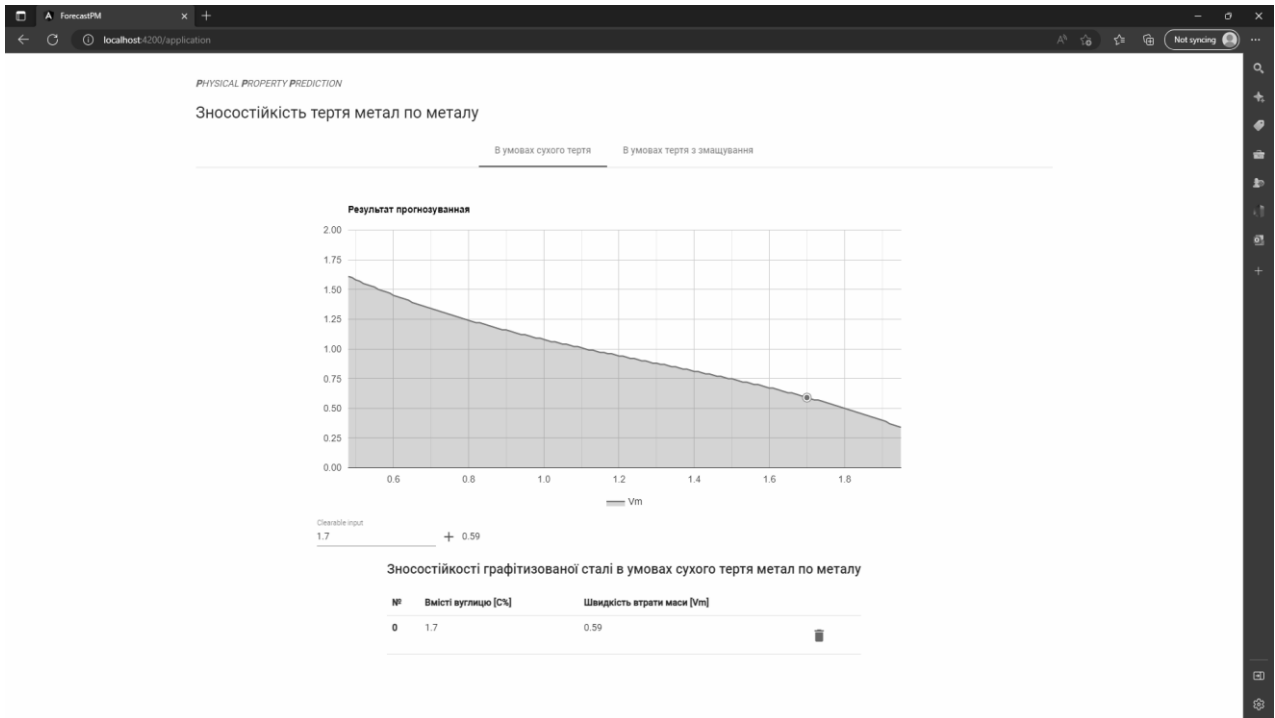


Рисунок 4.10 – Прогноз точок в додатку

Висновок: з вимірювань зносостійкості зразків ГС в умовах сухого тертя та тертя з змащування метал по металу, отриманих експериментальним шляхом, погрішність прогнозу додатку в допустимих нормах.

ВИСНОВКИ

Виходячи з результатів роботи, можна зробити наступні висновки. Базуючись на результатах експериментів, було складено математичну модель і, спираючись на неї, розроблено додаток, похибка якого не виходить за межі допустимої експериментальної похибки.

Собівартість при використанні додатка для прогнозування властивостей у виробництві деталей, значно менше, ніж при використанні традиційного підходу. Оскільки при зміні складу нема необхідності проводити експеримент, а досить лише ввести хімічний склад в додаток і отримати результат.

ПЕРЕЛІК ПОСИЛАНЬ

1. Forecasting Methods. URL: <https://www.wallstreetmojo.com/forecasting-methods/> (дата звернення: 18.10.2022).
2. Архітектури та технології WEB-служб. URL: http://eir.zntu.edu.ua/bitstream/123456789/2852/1/Stepanenko_Summary_of_1_ectures.pdf (дата звернення: 18.10.2022).
3. I-Factor. URL: <https://i.factor.ua/ukr/journals/nibu/2016/april/issue-32/article-17156.html> (дата звернення: 18.10.2022).
4. JMatPro Material Property Optimiser (МПО). URL: <https://www.sentesoftware.co.uk/jmatpro-mpo> (дата звернення: 18.10.2022).
5. SOLIDWORKS Materials – SOLIDWORKS Help. URL: <https://help.solidworks.com/HelpProducts.aspx> (дата звернення: 18.10.2022).
6. Ansys Granta MI. URL: <https://www.ansys.com/products/materials/granta-mi> (дата звернення: 18.10.2022).
7. Angular. URL: <https://github.com/angular/angular> (дата звернення: 22.11.2022).
8. Node.js. URL: <https://github.com/nodejs/node> (дата звернення: 22.11.2022).
9. Introduction to TypeScript and It's Features. URL: <https://www.htmlgoodies.com/javascript/introduction-to-typescript-and-its-features> (дата звернення: 22.11.2022).
10. Using Google Charts. URL: <https://developers.google.com/chart/interactive/docs> (дата звернення: 22.11.2022).

ДОДАТОК А

Лістинг програмного коду

A.1 HTML

```

<div class="workspace">
<h1 style="padding-top: 1rem">Зносостійкість тертя метал по металу</h1>
<mat-tab-group(selectedTabChange)="onTab($event)"                mat-align-
tabs="center">
<mat-tab label="В умовах сухого тертя">
<div class="view">
<div class="view__left_side">
<google-chart
    #chart
    (select)="onSelect($event)"
    [columns]="chartColumns"
    [data]="data"
    [options]="options"
    [type]="type"
    title="Test"
>
</google-chart>
<mat-form-field class="example-form-field">
<mat-label>Clearable input</mat-label>
<input [(ngModel)]="value" matInput max="1.95" min="0.48" step=".01"
type="number">
</mat-form-field>
<button (click)="addResult(value, currentValue, currentType)"
    aria-label="Example icon button with a heart icon"

```

```

        color="primary"
        mat-icon-button>
<mat-icon>add</mat-icon>
</button>
</span>{{ currentValue }}</span>
</div>
<div class="view__right_side">
<div *ngIf="resultsList[currentType] && resultsList[currentType].length"
class="user-list">
<h2>{{ tableTitle() }}</h2>
<table class="table table-condensed">
<thead>
<tr>
<th>№</th>
<th>Вмісті вуглицю [C%]</th>
<th>Швидкість втрати маси [Vm]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr *ngFor="let result of resultsList[currentType]; let i = index">
<th>{{ i }}</th>
<td>{{ result[0] }}</td>
<td>{{ result[1] }}</td>
<td>
<button (click)="deleteResult(i, currentType)" aria-label="Example icon button
with a heart icon"
        color="warn"
        mat-icon-button>
<mat-icon>delete</mat-icon>

```

```

</button>
</td>
</tr>
</tbody>
</table>
</div>
</div>
</div>
</mat-tab>
<mat-tab label="В умовах тертя з змашування">
<div class="view">
<div class="view__left_side">
<google-chart
  #chart
  (select)="onSelect($event)"
  [columns]="chartColumns"
  [data]="data"
  [options]="options"
  [type]="type"
  title="Test"
>
</google-chart>
<mat-form-field class="example-form-field">
<mat-label>Clearable input</mat-label>
<input [(ngModel)]="value" matInput max="1.95" min="0.48" step=".01"
type="number">
</mat-form-field>
<button (click)="addResult(value, currentValue, currentType)"
  aria-label="Example icon button with a heart icon"
  color="primary"

```

```

        mat-icon-button>
<mat-icon>add</mat-icon>
</button>
<span>{{ currentValue }}</span>
</div>
<div class="view__right_side">
<div *ngIf="resultsList[currentType] && resultsList[currentType].length"
class="user-list">
<h2>{{ tableTitle() }}</h2>
<table class="table table-condensed">
<thead>
<tr>
<th>№</th>
<th>Вмісті вуглицю [C%]</th>
<th>Швидкість втрати маси [Vm]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr *ngFor="let result of resultsList[currentType]; let i = index">
<th>{{ i }}</th>
<td>{{ result[0] }}</td>
<td>{{ result[1] }}</td>
<td>
<button (click)="deleteResult(i, currentType)" aria-label="Example icon button
with a heart icon"
        color="warn"
        mat-icon-button>
<mat-icon>delete</mat-icon>
</button>

```

```

</td>
</tr>
</tbody>
</table>
</div>
<br>

</div>
</div>
</mat-tab>
<span></span>
</mat-tab-group>
</div>

```

A.2 TYPESCRIPT

```

import { AfterViewInit, Component, ElementRef, OnInit, ViewChild } from
 '@angular/core';
import { MatTabChangeEvent } from '@angular/material/tabs';
import { ChartSelectionChangedEvent, ChartType } from 'angular-google-
charts';
import { find, first, get, last, round } from 'lodash';
import { BehaviorSubject } from 'rxjs';
import { DATA_VALUES, STEP } from './application.constant';

declare var google: any;

@Component({
  selector: 'app-application',
  templateUrl: './application.component.html',

```

```

styleUrls: ['./application.component.scss']
})
export class ApplicationComponent implements OnInit, AfterViewInit {

  @ViewChild('chart') chart?: ElementRef;

  type = ChartType.AreaChart;

  options = {
    title: 'Результат прогнозування',
    curveType: 'function',
    legend: { position: 'bottom' },
    colors: ['#6c63ff'],
    width: 930,
    height: 500,
    chartArea: { 'width': '90%', 'height': '70%' }
  }
  data: any;

  resultsList = {
    'GS_DRY': [],
    'GS_WET': []
  };
  chartColumns = [{ type: 'number', title: 'C%' }, 'Vm',];
  currentType: 'GS_DRY' | 'GS_WET' = 'GS_DRY';
  #value: BehaviorSubject<number> = new BehaviorSubject<number>(0.48);

  constructor() { }

  get value(): number {

```

```
    return this.#value.getValue();
  }

  set value(value: number) {
    this.#value.next(value);
  }

  get currentValue(): number {
    return get(find(this.data, { 0: this.value }), 1, 0)
  }

  currentData(currentType: string) {
    if (currentType == 'GS_DRY') {
      this.data = this.getDataArray(DATA_VALUES.GS_DRY);
      console.log(this.data)
    } else {
      this.data = this.getDataArray(DATA_VALUES.GS_WET);
      console.log(this.data)
    }
    return this.data
  };

  ngAfterViewInit(): void {
    this.currentData('GS_DRY');
  }

  ngOnInit(): void {
  }

  lanPol(x: number, x_values: number[], y_values: number[], size: number) {
```

```

let lagrange_pol = 0;
let basics_pol;

for (let i = 0; i < size; i++) {
  basics_pol = 1;
  for (let j = 0; j < size; j++) {
    if (j == i) continue;
    basics_pol *= (x - x_values[j]) / (x_values[i] - x_values[j]);
  }
  lagrange_pol += basics_pol * y_values[i];
}
return lagrange_pol;
}

tableTitle() {
  if (this.currentType == 'GS_DRY') {
    return 'Зносостійкості графітизованої сталі в умовах сухого тертя метал
по металу'
  } else {
    return 'Зносостійкості графітизованої сталі в умовах тертя з змащування
метал по металу'
  }
};

getDataArray(values: any) {
  let dataArray = [];
  let size = values.x.length;
  // @ts-ignore
  for (let x: number = first(values.x); x <= last(values.x) + STEP; x += STEP) {
    const y = this.lanPol(x, values.x, values.y, size);
  }
}

```



```

    if (x !== 1) {
      dataArray.push([round(x, 2), round(y, 2)]);
    } else {
      dataArray.push([round(x, 2), round(y, 2)]);
    }
  }
}
return dataArray;
}

onSelect($event: ChartSelectionChangedEvent) {
  const index = $event.selection[0].row
  if (index) {
    if (this.data[index][0] && this.data[index][1]) {
      this.setResult(this.data[index][0], this.data[index][1], this.currentType)
    }
  }
}

onTab($event: MatTabChangeEvent) {
  if ($event.index === 0) {
    this.data = this.getDataArray(DATA_VALUES.GS_DRY);
    this.currentType = 'GS_DRY'
    console.log(this.data)
  } else {
    this.data = this.getDataArray(DATA_VALUES.GS_WET);
    this.currentType = 'GS_WET'
  }
}

addResult(value: number, currentValue: number, currentType: string) {

```

```
// @ts-ignore
if (this.resultsList[currentType]) {
  // @ts-ignore
  this.resultsList[currentType].push([value, currentValue])
}
}

deleteResult(index: number, currentType: string) {
  // @ts-ignore
  this.resultsList[currentType].splice(index, 1);
}
}
```