МІНІСТЕРСТВО ОСВІТИ НАУКИ УКРАЇНИ

СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРАТ «ПВНЗ «ЗІЕІТ»

Циклова комісія з інформаційних технологій

ДО ЗАХИСТУ ДОПУЩЕНА

Голова циклової комісії,

спеціаліст в/к

_____ С.О. Сабанов

ВИПУСКНА РОБОТА МОЛОШОГО СПЕЦІАЛІСТА СТВОРЕННЯ ВЕБЗАСТОСУНКУ ДЛЯ ІНФОГРАФІЧНОГО ПРОЕКТУВАННЯ НА БАЗІ ТЕХНОЛОГІЇ REACT

Виконав

ст. гр. ІПЗ – 118К9 _____ Д.В. Єрємєєв

Керівник

доцент _____ О.А. Жеребцов

Запоріжжя

2022

СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРАТ «ПВНЗ «ЗІЕІТ»

Циклова комісія з інформаційних технологій

ЗАТВЕРДЖУЮ
Голова циклової комісії,
спеціаліст в/к
_____ С.О. Сабанов
17 січня 2022 р.

З А В Д А Н Н Я
НА ВИПУСКНУ РОБОТУ МОЛОДШОГО СПЕЦІАЛІСТА

студенту гр. ІПЗ-118К9 ,
спеціальності 121 - «Інженерія програмного забезпечення»
Єрємєєву Данилу Віталійовичу

1.Тема: «Створення вебзастосунку для інфографічного проектування на базі технології React»
затверджена наказом № 09.2 – 19 від 04 березня 2022 р.
2. Термін здачі студентом закінченої роботи: 18 червня 2022 р.
3. Перелік питань, що підлягають розробці:

1. Провести огляд предметної області та аналітику проблеми.
2. Провести огляд та аналіз популярних аналогів, зробити висновки про вимоги до проекту.
3. Розглянути методи та засоби створення вебзастосунків та обрати направлення розробки.
4. Провести огляд стеку технологій для розробки проекту та зробити вибір на підставі вимог.
5. Розробити проектування архітектури веб-застосунку.
6. Створити відповідний застосунок, спираючись на отримані дані.

7. Проаналізувати отримані результати.

8. Оформити звіт за результатами роботи

4. Календарний графік підготовки випускної роботи молодшого спеціаліста

№ етапу	Зміст	Терміни виконання	Готовність по графіку %, підпис керівника	Підпис керівника про повну готовність етапу, дата
1	Формулювання (корегування) теми випускної роботи молодшого спеціаліста та збір практичного матеріалу за темою випускної роботи	17.01.22-26.02.22		
2	**І атестація** І розділ випускної роботи молодшого спеціаліста	28.03.22-02.04.22		
3	**ІІ атестація** ІІ розділ випускної роботи молодшого спеціаліста	10.05.22-14.05.22		
4	**ІІІ атестація** ІІІ розділ випускної роботи молодшого спеціаліста, висновки та рекомендації, додатки, реферат	30.05.22-04.06.22		
5	Перевірка випускної роботи молодшого спеціаліста програмою «Антиплагіат»	30.05.22-18.06.22		
6	Доопрацювання випускної роботи молодшого спеціаліста, підготовка презентації, отримання відгуку керівника і рецензії	06.06.22-11.06.22		
7	**Попередній захист випускної роботи молодшого спеціаліста**	**14.06.22-18.06.22**		
8	Подача випускної роботи молодшого спеціаліста на кафедру	за 3 дні до захисту		
9	**Захист випускної роботи молодшого спеціаліста**	**20.06.22-25.06.22**		

Керівник _____ О.А. Жеребцов
(підпис) (ініціали, прізвище)

«____» _____ 2022 р.

Завдання отримав до виконання _____ Д.В. Єрємєєв
(підпис студента) (ініціали, прізвище)

«____» _____ 2022 р.

РЕФЕРАТ

Випускна робота молодшого спеціаліста містить 92 сторінки, одна таблиця, 37 рисунків, один додаток, дев’ять лістингів, 26 бібліографічних посилань.

Метою роботи є створення веб-застосунку для інфографічного проектування на базі технології React.

Об’єктом дослідження є проектування взаємодій даних в вигляді діаграм та таблиць.

Предметом дослідження є веб-застосунок для графічного проектування.

Здійснено детальний огляд предметної області та сучасних аналогів.

Проект реалізовано за допомогою таких засобів, як HTML, CSS, TypeScript, React.

Веб застосунок є зручним в використанні та має гарний та сучасний інтерфейс. Додаток допомагає користувачу структурувати данні в зручному форматі. Це може заощаджувати час.

У результаті роботи був створений веб-додаток для інфографічного проектування даних. Отримані результати повністю задовольняють вимогам, які були поставлені перед замовником. Веб застосунок коректно відображається у всіх сучасних браузерах для будь-яких операційних систем.

REACT, HTML, CSS, TS, ІНФОГРАФІЧНЕ ПРОЕКТУВАННЯ, ВЕБ-ЗАСТОСУНОК

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 7

ВСТУП 9

РОЗДІЛ 1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ 10

1.1 Прикладна задача 10

1.2 Огляд проблематики 11

1.3 Огляд аналогічних додатків 15

1.3.1 web застосунок SmartDraw 15

1.3.2 web застосунок Creately 17

1.4 Вимоги базового функціоналу до програмного продукту 18

1.5 Висновки за розділом 19

РОЗДІЛ 2 ЗАСОБИ ТА МЕТОДОЛОГІЇ РОЗРОБКИ ДОДАТКУ 20

2.1 React 20

2.1.1 Різниця між SPA та MPA 23

2.1.2 Віртуальний DOM 26

2.1.3 Методи життєвого циклу 27

2.1.4 Класи проти функцій 28

2.1.5 Класи проти функцій 29

2.1.6 Переваги та недоліки React 30

2.1.7 SEO 31

2.2 Backend 32

2.2.1 .NET та мова програмування C# 32

2.2.2 Web API, Rest API 33

2.2.3 MS SQL Express 36

2.2.4 IIS Server 37

2.3 Мова програмування TypeScript та її сфери використання 37

2.4 React Syncfusion 40

2.5 Середа розробки WebStorm 42
2.6 Висновки за розділом 43

РОЗДІЛ 3 ПРОГРАМНИЙ МОДУЛЬ 44
3.1. Проектування системи 44
3.2 Програмування системи 47
3.2.1 Архітектура додатку 48
3.2.2 Головний компонент 55
3.2.3 Вкладка сумісність 59
3.2.4 Вкладка розкладу 63
3.3 Висновки за розділом 64

ВИСНОВКИ 65
ПЕРЕЛІК ПОСИЛАНЬ 66
ДОДАТОК А 69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Скорочення	Повна назва	Пояснення/переклад
ASP.NET	Active Server Pages для .NET	Платформа розробки веб-додатків, до складу якої входять: веб-сервіси, програмна інфраструктура, модель програмування, від компанії Майкрософт
API	Application Programming Interface	Прикладний програмний інтерфейс
CDN	Content delivery network	Мережа доставки контенту
CSS	Cascading Style Sheets	Каскадні таблиці стилів
CSV	Comma - Separated Values	Значення, розділені комою
DOM	Document Object Model	Об'єктна модель документа
FTP	File Transfer Protocol	Протокол, призначений передачі файлів у комп’ютерній мережі
IIS	Internet Information Services	Пропрієтарний набір серверів для кількох онлайнових служб від компанії Microsoft.
ISV	International Standard Version	Незалежний постачальник ПЗ
IT	Information Technology	Система методів, процесів та способів використання обчислювальної техніки і систем зв'язку для створення, збору, передачі, пошуку, оброблення та поширення інформації з метою ефективної організації діяльності людей
JS	Java Script	Динамічна, об'єктно-орієнтована прототипна мова програмування; реалізація стандарту ECMAScript
JSON	JavaScript Object Notation	Текстовий формат обміну даними на основі JavaScript
JSX	JavaScript Syntax Extension	Розширення REST до синтаксису мови JavaScript, яке надає спосіб структурування відтворення компонентів за допомогою синтаксису, знайомого багатьом розробникам
HTML	HyperText Markup Language	Стандартизована мова розмітки документів у Всесвітньому павутинні
HTTP	Hyper Text Transfer Protocol	Протокол передачі гіпертекстових документів – протокол передачі даних, що використовується в комп'ютерних мережах
MPA	Multi Page Application	Програма, яка вимагає перезавантаження сторінки кожного разу, коли вміст сторінки змінюється

MVC	Model View Controller	Модель–представлення–контролер, архітектурний шаблон
MVP	Minimum viable product	Мінімально життєздатний продукт
.NET	.NET	кросплатформна реалізація для запуску веб-сайтів, служб та консольних програм у Windows, Linux та macOS
PDF	Portable Document Format	Формат файлу, створений і підтримуваний компанією Adobe Systems
PHP	Hypertext Preprocessor	Препроцесор гіпертексту
REST	Representational State Transfer	Архітектурний стиль взаємодії компонентів розподіленого застосування в мережі
SEO	Search Engine Optimization	Комплекс заходів щодо збільшення видимості сайту в пошукових системах за цільовими пошуковими запитами
SOAP	Simple Object Access Protocol	Протокол обміну повідомленнями у розподіленому обчислювальному середовищі
SPA	Single Page Application	Програма, яка працює всередині браузера і не потребує перезавантаження сторінок під час його завантаження
SQL	Structured query language	Мова структурованих запитів
TLS	Transport layer security	Протокол захисту транспортного рівня
UI	User Interface	Користувацький інтерфейс
URI	Uniform Resource Locator	Уніфікований локатор ресурсів або адреса ресурсу
URL	Uniform Resource Locator	Уніфікований локатор ресурсів або адреса ресурсу
WCF	Windows Communication Foundation	Програмний фреймворк, який використовується для обміну даних між додатками та входить до складу .NET Framework
XHTML	Extensible hypertext mar kup language	Сімейство мов розмітки веб-сторінок на основі XML, що повторюють та розширюють можливості XHTML
XML	Extensible Markup Language	Розширювана мова розмітки
XML-RPC	Extensible Markup Language Remote Procedure Call	Базується на XML стандарті (протоколі) виклику віддалених процедур, вирізняється простотою використання
Веб	Web	Інтернет-простір
ПЗ	Програмне забезпечення	Сукупність програм системи оброблення інформації та програмних документів
СУБД	Система управління базами даних	Поєднання ПЗ та лінгвістичних інструментів для загальних або спеціальних цілей, що забезпечують управління створенням та використанням баз даних

ВСТУП

Наразі в Україні все більш популярними стають новітні технології серед користувачів. Ці технології полегшують життя всім користувачам з точки зору швидкості та якості обслуговування.

Метою випускної роботи є створення веб-застосунку, реалізація якого буде здійснена за допомогою сучасного React. Основною ідеєю є створення зручного і гарного веб-застосунку для покрокового розкладу того як буде відбуватися процес міграції даних, у вигляді діаграм і таблиць.

Міграція даних - це процес переміщення великих обсягів даних з одного місця до іншого. Це дуже кропітка робота бо нам необхідно тримати в голові великий обсяг інформації. Зазвичай план міграції записують на листку або в Word, Excel. Цей спосіб не є зручним. Задля того щоб цей процес полегшити, і було вигадано замовником таку концепцію.

Застосунок має унікальну ідеє, та призначений для внутрішнього використання в компанії замовника, яка знаходиться в Австралії.

Під час виконання завдання здобуваються навички роботи в команді, правильного планування часу та виконання комерційних задач.

РОЗДІЛ 1
ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Прикладна задача

Веб-застосунками, або веб-системами - це різні програмні продукти, доступ до яких здійснюється через веб-інтерфейс. Розробка веб-застосунків потрібна наприклад різним комерційним організаціям, тому функціонал веб - застосунків зазвичай включає потужні бізнес-орієнтовані інструменти.

Сучасні веб-застосунки володіють наступними властивостями:

- масштабованість веб-системи;
- інтеграція з іншими системами;
- розмежування прав доступу до різного функціоналу;
- зручне розгортання та обслуговування системи.

У новинах постійно з’являється інформація про успішні кібератаки. Тому безпека застосунку повинна бути на першому місці. Кажуть що не немає такого сайту який неможливо хакнути, але необхідно прагнути забезпечити максимальну безпеку. Зневага до цього може призвести до витоку персональних даних користувачів [1].

Безпека веб-застосунків це розділ інформаційної безпеки який відповідає за вирішення питань безпеки веб-сайтів, веб-застосунків та веб-сервісів.

З іншого боку, будь-який веб-розробник і пересічний користувач, що просто цікавиться цією темою, повинен розуміти основну різницю між веб додатком і веб-сайтом.

Веб сайти виділяються насамперед своєю високою інформативністю. Головне завдання веб-сайту - донести щось до відвідувача, при цьому взаємодія з боку користувача досить обмежена.

Веб сайти здебільшого статичні, і інформація, яку бачить один користувач буде абсолютно ідентичною, як якщо її перегляне інший.

1.2 Огляд проблематики

Комп'ютерні та інформаційні технології дедалі більше проникають у життя людини, і вже давно стали невід'ємною частиною економіки, суспільства, політичної та культурної сфер.

Україна займає почесне 11-те місце за рейтингом кращих розробників в світі [10]. Це означає, що українські розробники користуються попитом в світі (див. рис.1.1).

Rank	Country	Score Index
1	China	100.0
2	Russia	99.9
3	Poland	98.0
4	Switzerland	97.9
5	Hungary	93.9
6	Japan	92.1
7	Taiwan	91.2
8	France	91.2
9	Czech Republic	90.7
10	Italy	90.2
11	Ukraine	88.7
12	Bulgaria	87.2
13	Singapore	87.1
14	Germany	84.3
15	Finland	84.3
16	Belgium	84.1
17	Hong Kong	83.6
18	Spain	83.4
19	Australia	83.2
20	Romania	81.9
21	Canada	81.7
22	South Korea	81.7
23	Vietnam	81.1
24	Greece	80.8
25	Sweden	79.9

Rank	Country	Score Index
26	Netherlands	78.9
27	Chile	78.4
28	United States	78.0
29	United Kingdom	77.7
30	Turkey	77.5
31	India	76.0
32	Ireland	75.9
33	Mexico	75.7
34	Denmark	75.6
35	Israel	74.8
36	Norway	74.6
37	Portugal	74.2
38	Brazil	73.4
39	Argentina	72.1
40	Indonesia	71.8
41	New Zealand	71.6
42	Egypt	69.3
43	South Africa	68.3
44	Bangladesh	67.8
45	Colombia	66.0
46	Philippines	63.8
47	Malaysia	61.8
48	Nigeria	61.3
49	Sri Lanka	60.4
50	Pakistan	57.4

Рис. 1.1 – Рейтинг якості розробників в світі

На мою думку, та багато хто так вважають, що саме IT -сфера є одною з найвигіднішою для заробітку. Професія «програміст» одна з найдорожчих та затребуваних у світі. Особливо цінуються фахівці, здатні працювати у різних сферах, брати участь у проектах, створювати коди, ПЗ, автоматизувати виробництво, впроваджувати цифрові технології, робити математичні та інформаційні моделі, макети. Найзатребуванішими фахівцями у 2021 році були програмісти. У цій сфері було 41,8% проектів [9]. Ще шукали дизайнерів (19,8% замовлень) та тих, хто працює з текстами (13,5% замовлень). Раніше, IT-сфера довгий час була недооцінена в нашій країні з позиції її впливу на формування потенціалу зростання національної економіки, відбувалось практично невтручання держави в розвиток саме цієї сфери, але великий ринок збуту зі зростаючими гарних характеристик дозволили сформувати потужну платформу для розвитку IT-сфера [2].

Головний критерій, що визначає рівень заробітної плати програміста у США чи Європі, – знання мови програмування. Також прибутки залежать від місця роботи. Якщо фахівець працює у великій міжнародній компанії або корпорації, то заробіток у нього буде більшим, ніж у програміста, зайнятого у невеликій приватній фірмі. Заробіток українського спеціаліста залежить від виду роботи. Статистика говорить, що платіжна система Payoneer, в середньому заробляють $650 - $2100 на місяць. Середній дохід у період з 2019 до 2021 року зріс на 18%, загальні зарахування – на 51%.

В України широко представлені сервісні компанії, що спеціалізуються на розробці, тестуванні і супроводі програмного забезпечення на замовлення, і працюють не тільки на внутрішньому ринку, але і на ринках Європи, США та інших країн. За даними Міжнародної асоціації IAOP, тринадцять як українських IT-компаній, так і міжнародних корпорацій з великими представництвами в Україні, увійшли в рейтинг 100 кращих аутсорсингових компаній світу: EPAM Systems, Luxoft, N-iX, Miratech, ELEKS, Intetics Inc,

SoftServe, Softjourn Inc, Ciklum, Sigma Software, TEAM International Services Inc, Program-Ace та Softengi21.

Україна має велику кількість висококваліфікованих спеціалістів. Майже 2/3 українських спеціалістів мають вищу освіту і високі знання англійської мови.

Нажаль рівень життя та заробітку в Україні залишається низькими, а безробіття – високий [7]. На це вплинуло економічне зростання і політична ситуація 2013-2014 років. Спалах військових дій у східній частині країни в 2014 році вплинули на ринок праці в країні, мотивувавши все більше і більше людей шукати роботу в IT-сфері. Насамперед із за можливості працювати дистанційно, це має ряд переваг, а саме в пандемію і через війну яка розпочалася с 24 лютого 2022року. В карантин дуже багато людей втратило роботу, а на IT-сферу майже не вплинуло, тому що саме це зіграло ключову роль, що запустило нову хвилю інтересу до віддаленої роботі не тільки нашої країни, а всій планети.

Зараз сфера IT в Україні дуже стрімко розвивається. На 28% за рік збільшився середній прибуток українських IT-компаній, які працюють із іноземними клієнтами [8]. За липень-вересень 2021 року в середньому прибуток складає $4390. Разом із тим збільшилася і кількість самих компаній – на 7% з третього кварталу 2020 року до третього кварталу 2021 року (див. рис. 1.2).

У США програмісти отримують більше, ніж фахівці в аналогічних сферах у Європейському Союзі чи СНД. Кількість вакансій там постійно зростає, потрібно лише відповідати певним критеріям, мати потрібну кваліфікацію, знати англійську мову.

В Америці знаходяться такі компанії-монстри, як Google, Apple, Facebook, 3D Systems та інші. Робота в таких компаніях стане чудовим етапом у розвитку кар'єри. Робота програмістом у США робить резюме привабливим для наймачів, підвищує шанси отримати грін-карту.

Рисунок 1.2 – Заробіток в Україні з III кварталу 2019 року по III квартал 2021 рік.

Вони повинні володіти мовами програмування, як Java, C++, CSS, PHP, JavaScript. Програмісти у США, як і в інших країнах світу, одержують зарплату за годину роботи. Так, 1 година праці з вакансії java developer коштує від 40 до 70 $.

Junior спеціалісти в Україні в середньому розраховують на заробітну плату в розмірі 1000-1500 $ на місяць, той же час, як у США в середньому розраховують на прибуток близько 6200 доларів, але якість наданих IT-послуг в Україні не гірше, а ціна, навіть з урахуванням щорічного зростання заробітної плати в цій сфері нижче [5]. Всі ці дані наведені в таблиці 1.1.

Пов’язано це, тільки з досить низькою мінімальною заробітною платою націй країні.

Таблиця 1.1 – Середня заробітна плата програмістів 2022 г.

№ п/п	Країни	зарплата в місяць
1	Україна	от 1,0 до 1,5 тис. $
2	Білорусія	от 1,0 до 1,5 тис. $
3	Чехія	от 2,0 тис. $
4	Германії	2,6 тис. $
5	Канада	4,3 тис. $
6	Великобританія	от 5,2 тис. $
7	США	от 6,2 тис. $
8	Швейцарії	от 11,3 тис. $

Тому, співпраця іноземних клієнтів з українськими спеціалістами є вигідною як для однієї, так і для іншої сторони.

Для написання веб - застосунку, важливо вибрати стек технологій.

1.3 Огляд аналогічних додатків

На даний момент працюючих аналогів застосунку за концепцією замовника не існує. Існують лише схожі за ідеєю та дизайном застосунки.

Розглянемо найбільш популярні рішення за схожею ідеєю.

1.3.1 Вебзастосунок SmartDraw

SmartDraw – це інструмент діаграм, що використовується для створення потокових діаграм, організаційних діаграм, інтелектуальних карт, діаграм

проектів та інших бізнес-візуальних елементів [23]. Як це показано на рисунку 1.3:

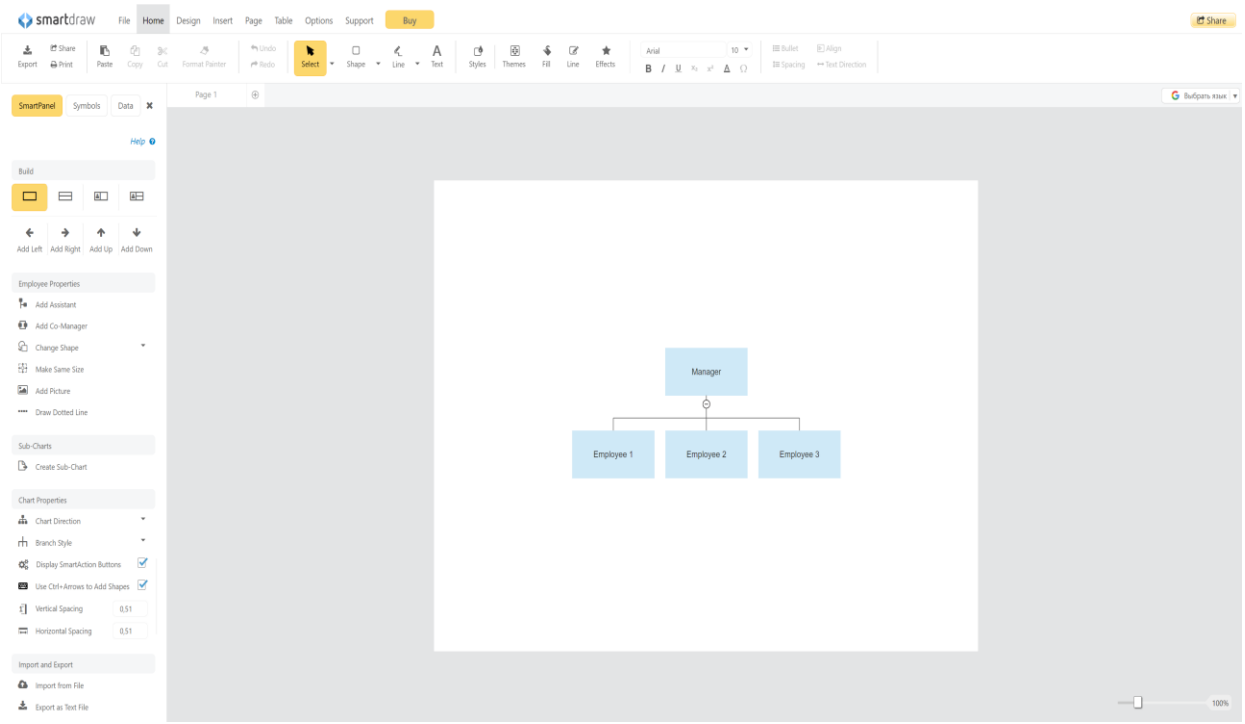


Рисунок 1.3 – Вигляд web застосунку SmartDraw

Має особливості:

- Зручний інтерфейс
- Гарний дизайн
- Зручне створення діаграм
- Великий вибір опцій
- Різноманітність фігур для створення діаграм.

SmartDraw, найбільш відома створенням діаграм для бізнесу, таких як блок-схеми та організаційні діаграми, а також простим перетягуванням планів поверхів та схем, перезапустила настільний додаток, який додає функціональні можливості та, за словами компанії, підвищує його користувальницьку привабливість, дружелюбність у порівнянні з онлайн-версією.

1.3.2 Вебзастосунок Creately

Creately – це інструмент візуальної спільної роботи SaaS з можливостями побудови діаграм та дизайну, розроблений Cinergix (див. рис.1.4)

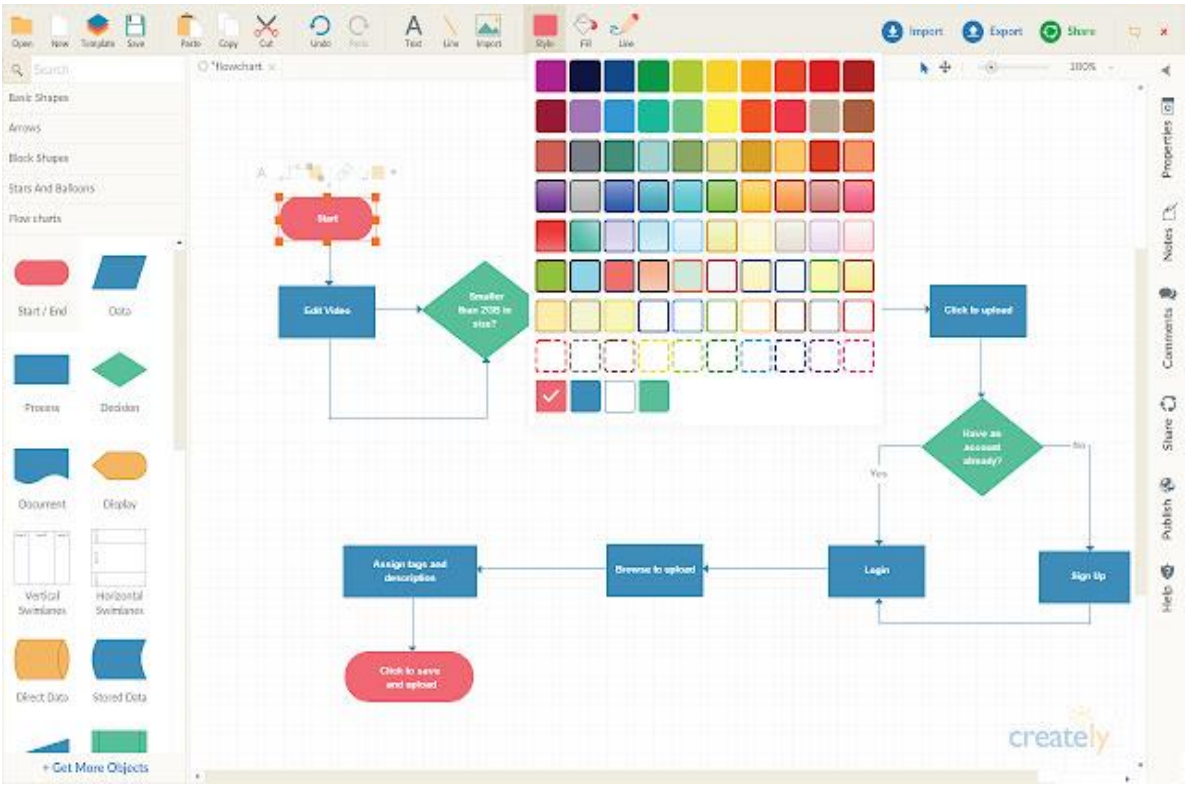


Рисунок 1.4 – Вигляд web-застосунок Creately [14]

SaaS – одна з форм хмарних обчислень, модель обслуговування, за якої передплатникам надається готове прикладне ПЗ, яке повністю обслуговує провайдер.

Має особливості:

- Різноманітність можливостей
- Існує на всіх платформах
- Цікавий дизайн
- Можливість імпорту та експорту діаграм

Цей додаток дозволяє створювати прості та складні логічні елементи - від анотованих електричних схем до цифрових та аналогових логічних проектів.

Найбільшими недоліками веб-застосунку є відсутність аналітичних можливостей над отриманими схемами та потреба у запуску FLASH в браузері. Як відомо – ця технологія є застарілою в наш час і містить велику кількість вразливостей, якими можуть скористатись зловмисники. Крім того сервіс потребує авторизації та дуже повільно працює.

1.4. Вимоги до програмного продукту

Вимоги до майбутнього продукту:

1. Наявність функціоналу у вигляді менеджменту проектів застосунку, тобто створення проекту, надання імені, видалення та перейменування.
2. Наявність вкладок: Process, Compatibility, Lists, Schedule.
3. Надання користувачеві зручну можливість створення схем в виді діаграм, та наповнювати блоки діаграми інформацією. Блок повинен складатися з назви, статусу, текстового поля та таблиці.
4. Наявність збереження проектів до бази даних.
5. Надання користувачеві зрозумілий, гарний та зручний сервіс

Виходячи з представлених вимог, щодо функціоналу майбутнього веб-застосунку було здійснено:

1. Менеджмент проектів відповідно до вимог. Можливість роботи з проектами: створювати редагувати, переключатися, видаляти.
2. Головна сторінка, де ми можемо вибрати наступні вкладки: Process, Compatibility, Lists, Schedule.

- В вкладці Process відбувається побудова плану в вигляді діаграм, де кожний блок містить назву, статус, опис та таблицю.

- Вкладка Compatibility містить таблицю.
- В кладці Lists знаходиться чотири таблиці які пов’язані між собою спільними колонками.
- Schedule це вкладка в якій знаходиться текст для копіювання.

3. Наявність контекстних меню.
4. Зручний та зрозумілий веб-застосунок який економить час.
5. Усі дії користувача зберігаються на локальному сервері.
6. Маємо можливість створювати стільки проектів та діаграм, скільки хочемо, бо обмежень немає.

1.5. Висновки за розділом

Було здійснено огляд щодо IT індустрії в Україні. Установлено, що українські спеціалісти користуються попитом у світі.

Виявлено вимоги базового функціоналу до програмного продукту.

Здійснено порівняльна характеристика схожих за оглядом додатків, та встановлено їх переваги та можливі недоліки.

Було висунуто вимоги до програмного продукту з метою створення зручного і гарного веб-застосунку для покрокового розкладу того як буде відбуватися процес міграції даних, у вигляді діаграм і таблиць.

РОЗДІЛ 2
ЗАСОБИ ТА МЕТОДОЛОГІЇ РОЗРОБКИ ДОДАТКУ

2.1 React

React - одна з найбільш простих JavaScript бібліотек з відкритим кодом. Він використовується для побудови інтерактивних інтерфейсів користувача. Це ефективна, декларативна та гнучка бібліотека. Це не цілий каркас, а лише бібліотека фронтових партій (рис. 2.1). Це дозволяє створювати або створювати складні користувацькі інтерфейси, використовуючи ізольовані та невеликі шматочки коду, відомі як компоненти.

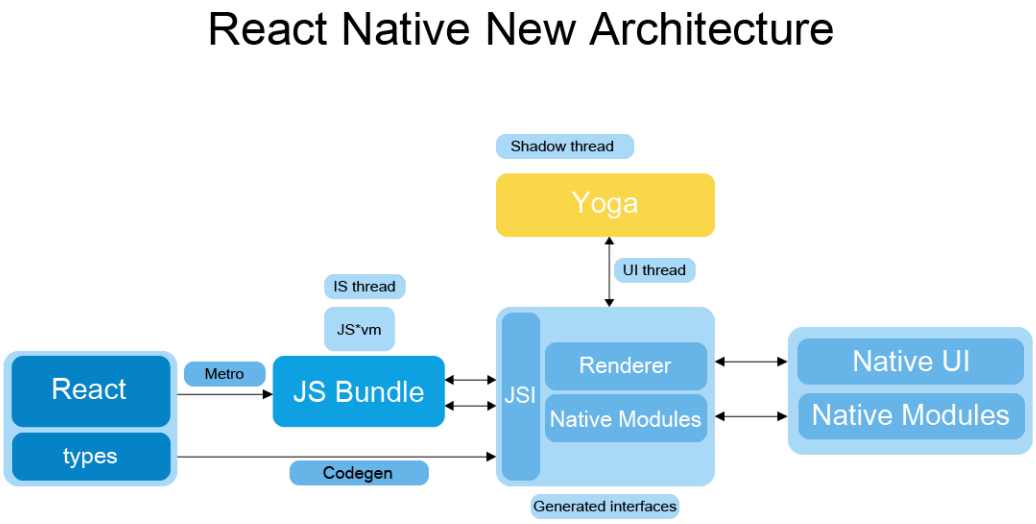


Рисунок 2.1 – Архітектура бібліотеки React

В основі будь-якого веб-сайту лежить мова гіпертекстової розмітки HTML, каскадні сторінки стилів CSS та JavaScript. Веб-сайт для свого функціонування не вимагає знання мов програмування або вміння роботи з СУБД [17].

HTML виступає каркасом всього сайту, він «пояснює» браузеру, де і які елементи хоче бачити на сторінці розробник, а так само як браузеру потрібно їх інтерпретувати. HTML не є мовою програмування, тому що вона не виконує обробку даних, а лише їх відображення.

CSS налаштовує дизайн сайту. Так би мовити, «одягає» поверх елементів сторінки певні вбрання.

JavaScript – це вже мова програмування, яка використовується для створення динамічних скриптів на веб-сторінці.

В контексті веб-сайту, JS дає набір можливостей для аналізу вмісту сторінки, його зміни, відстеження взаємодій користувача зі сторінкою та проведення реакцій на них (див. рис. 2.2).

Рисунок 2.2 – Складові веб-сайту/фронтенду веб-додатку

Зараз, JavaScript виконуватися не тільки в браузері, а і на сервері чи на іншому пристрої, який має спеціальну програму, що називається "движком" JavaScript.

Веб-сайти статичні, обмежені у функціоналі та інтерактивності, не надають можливості передавати дані, зберігати їх та маніпулювати ними. Але при цьому прості та швидкі у розробці, а також є менш ресурсозатратними.

Веб сайти підходять для певних цілей, проте явно досить часто даного функціоналу недостатньо для повноцінної роботи [4].

Веб-застосунок, на відміну від веб-сайту, дає можливість не тільки отримувати статичну, але й динамічну інформацію, а також віддавати її та маніпулювати інформацією на сторінці, як це показано на рисунку 2.3.

Веб-застосунок є інтерактивним, і дозволяє, наприклад: зареєструватися, відправити коментар, оформити замовлення, зберегти якісь дані та отримати до них доступ пізніше.

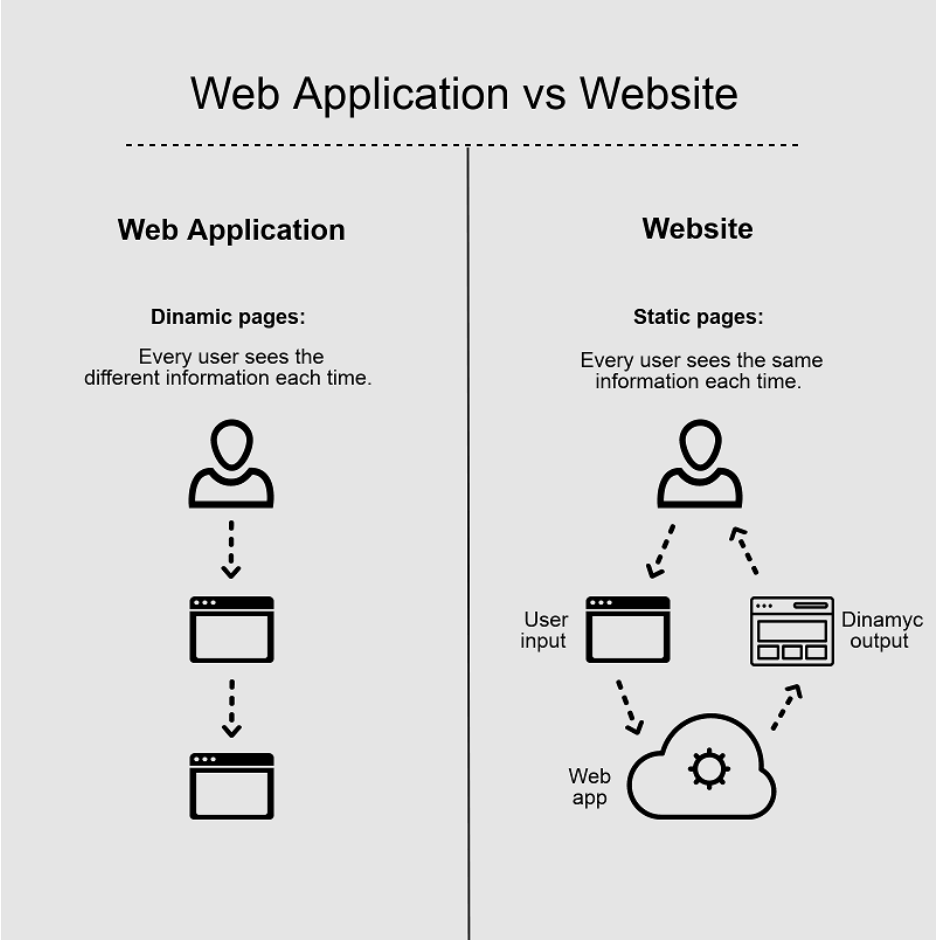


Рисунок 2.3 – Порівняння веб-сайту і веб-застосунку

Веб-застосунок – це в першу чергу клієнт серверний додаток, основна частина якого міститься на віддаленому сервері, а користувач отримує доступ

до нього, використовуючи інтерфейс (UI) через веб браузер з активним мережевим підключенням.

Сторінки веб-додатків формуються динамічним шляхом, так би мовити «на льоту», залежно від інформації, що надійшла з сервера.

Веб-програми мають тісний зв'язок із сервером, надсилаючи на нього безліч запитів. Вони не завжди мають інформативний зміст, як веб-сайти, а використовуються для виконання конкретних завдань – підрахунків, статистики, перекладів, конвертації, обміну повідомленнями та/або файлами.

Мета React полягає в тому, щоб бути простим, швидким та масштабованим. Ця бібліотека дозволяє розробникам створювати складні та великі веб-застосунки, котрі при оновленні інформації на сторінці, не оновлюють цю сторінку (SPA).

Netflix, Khan, Sony, Academy, Yahoo, Airbnb, Atlassian та інші. використовують React.

React.js — бібліотека від Facebook, являється домінуючою серед засобів для побудови SPA. Тобто створена для вирішення питання з частковим оновленням вмісту сторінки при розробці SPA. Гарно підходить для створення великих веб-застосунків так простих застосунків.

2.1.1 Різниця між SPA та MPA

Серед можливих архітектур веб-додатків, на сьогоднішній день, найбільш поширені односторінкові програми (SPA) або багатосторінкові програми(MPA) (рис 2.4)[19].

Односторінкова програма (SPA) - це програма, яка працює всередині браузера і не вимагає перезавантаження сторінки під час використання. SPA запитує розмітку та дані незалежно та відображає сторінки прямо у браузері. Ми можемо зробити це завдяки просунутим фреймворкам JavaScript, таким як React, Angular, Ember, Meteor, Knockout

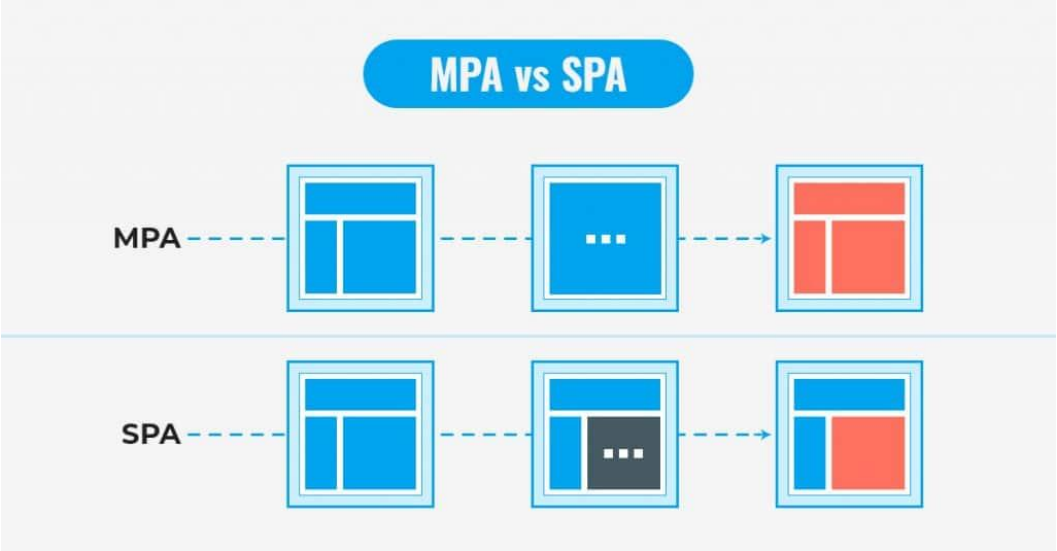


Рисунок 2.4 – Порівняння веб-сайту і веб-застосунку

Плюси односторінкового додатку:

- SPA працює швидко, оскільки більшість ресурсів (HTML+CSS+Scripts) завантажуються лише один раз протягом усього життєвого циклу програми.
- Розробка спрощена та оптимізована.
- SPA легко налагоджувати за допомогою Chrome, тому що ви можете відстежувати мережеві операції, досліджувати елементи сторінки та пов'язані з нею дані.
- Зробити мобільний додаток простіше, тому що розробник може повторно використовувати той самий внутрішній код для веб-програми та власного мобільного додатка.
- SPA може ефективно кешувати будь-яке локальне сховище.

Мінуси односторінкової програми:

- Зробити SEO-оптимізацію односторінкового додатка дуже складно.
- Він завантажується повільно, тому що для завантаження клієнта потрібні важкі фреймворки клієнта.

- Для цього потрібно, щоб JavaScript був присутній та включений.
- У порівнянні з «традиційним» програмою SPA менш безпечний.
- Витік пам'яті JavaScript може навіть уповільнити роботу потужної системи.

Багатосторінкові програми (MPA) працюють «традиційним» способом. Кожна зміна, наприклад, відображати дані або надсилати дані назад на запити сервера, що відображають нову сторінку сервера в браузері. Ці програми більші, більше, ніж SPA, тому що вони повинні бути такими. Через велику кількість контенту ці програми мають багаторівневий інтерфейс користувача.

Плюси багатосторінкового додатку:

- Це ідеальний підхід для користувачів, яким потрібна наочна мапа того, куди йти в додатку.
- Дуже добре і легко для правильного управління SEO.

Мінуси багатосторінкової програми:

- Немає можливості використовувати один і той же сервер з мобільними програмами.
- Фронтенд і бекенд ці технології тісно пов'язані.
- Розвиток стає досить складним. Розробнику необхідно використовувати фреймворки як клієнтської, так серверної частини. Це призводить до збільшення часу розробки програми.

Очевидна та найбільш помітна різниця між ними – швидкість. Перевага SPA полягає в тому, що вони попередньо завантажені, і хоча вони зазвичай мають деякі початкові затримки під час вилучення даних, наступні запити виконуються набагато швидше через кеш.

Однак, в односторінковому застосунку, порівняно з багатосторінковим додатком MPA, потрібно запитувати дані для кожної сторінки, до якої звертається користувач, і мають набагато повільніші програми.

2.1.2 Віртуальний DOM

React має підтримку віртуального DOM. Він не покладається тільки на DOM браузера. Це дозволяє бібліотеці визначити, які частини DOM змінилися, порівняно зі збереженою версією віртуального DOM, і таким чином визначити, як найефективніше оновити DOM браузера. Бібліотека сама вирішує які компоненті зі сторінки потребують оновлення.

JavaScript працює з деревом DOM, як це видно з рисунку 2.5. Яке своєю чергою є об'єктною моделлю документа. Коли браузер отримує вихідний HTML документ, він розбирає його на об'єкти і вже на їх основі будує DOM дерево. І лише потім описує його на сторінці.

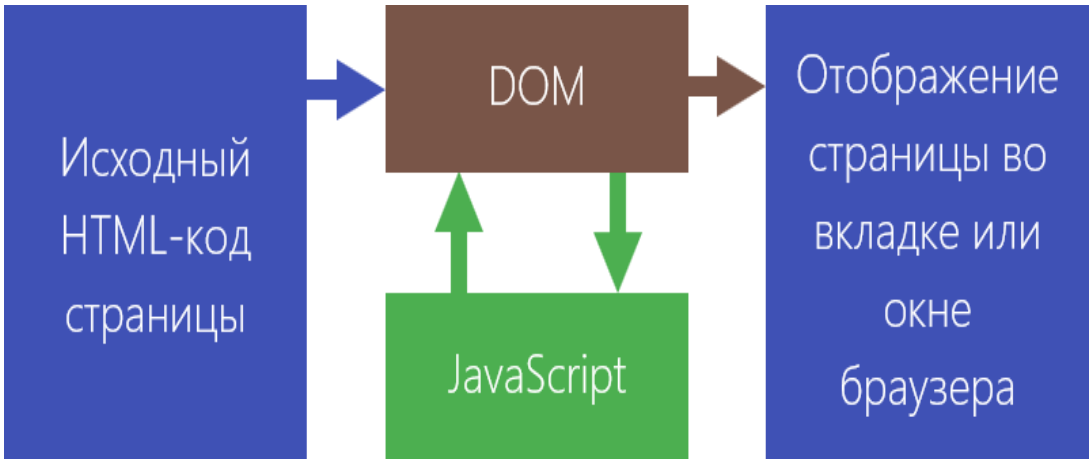


Рисунок 2.5 – Схема роботи JS з DOM деревом

DOM дерево складається з нодів (вузлів). Будь-який вузол – це сам елемент, його атрибути та текстова нода. Усі вузли пов'язані відносинами батько-дитина.

І саме існування DOM дерева дозволяє JS робити зміни на сторінці та отримувати дані про її елементи.

2.1.3 Методи життєвого циклу

Методи життєвого циклу — це різні методи, які вбудовуються за допомогою ReactJS [3]. Вони дозволяють розробнику обробляти дані в різних точках життєвого циклу програми React (рис. 2.6).

У процесі роботи компонент проходить через низку етапів життєвого циклу. На кожному з етапів викликається певна функція, в якій ми можемо визначити будь-які дії:

- constructor (props): конструктор, у якому відбувається початкова ініціалізація компонента,
- static getDerivedStateFromProps(props, state): викликається безпосередньо перед рендерингом компонента. Цей метод не має доступу до поточного об'єкта компонента і повинен повертати об'єкт для оновлення об'єкта state або значення null, якщо нічого оновлювати,
- render(): рендеринг компонента,
- componentDidMount(): викликається після рендерингу компонента. Тут можна виконувати запити до віддалених ресурсів,

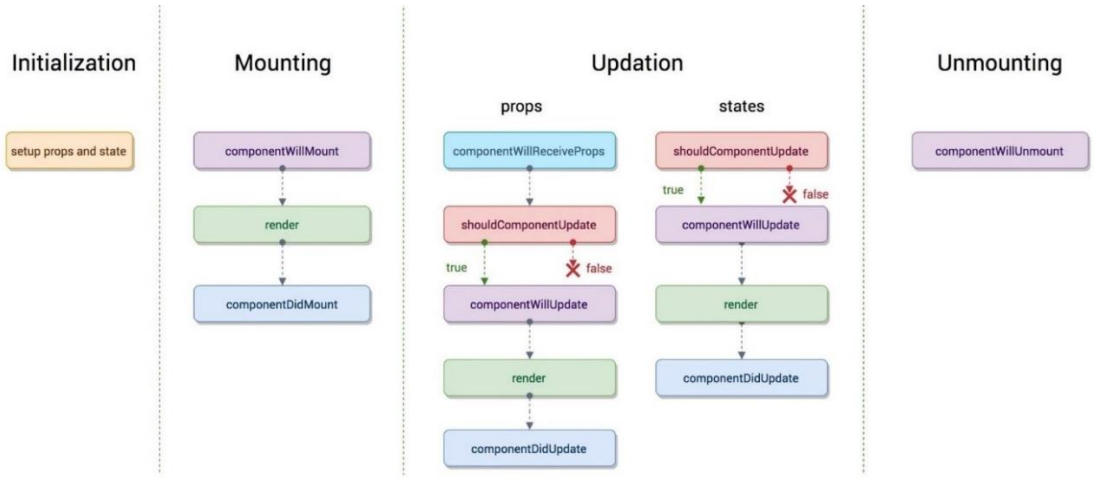


Рисунок 2.6 – Зміни життєвого циклу React

- componentWillUnmount(): викликається перед видаленням компонента з DOM.

Крім основних етапів чи подій життєвого циклу, є ряд функцій, які викликаються при оновленні стану після початкового рендерингу компонента, якщо в компоненті відбуваються оновлення:

- shouldComponentUpdate (nextProps, nextState): викликається щоразу під час оновлення об'єкта props або state. Як параметр передаються новий об'єкт props і state.
- getSnapshotBeforeUpdate (prevProps, prevState): викликається безпосередньо перед компонентом. Він дозволяє компоненту отримати інформацію із DOM перед можливим оновленням. Повертає як значення якийсь окремий аспект, який передається як третій параметр у метод componentDidUpdate() і може враховуватися в componentDidUpdate при оновленні.
- componentDidUpdate (prevProps, prevState, snapshot): викликається відразу після оновлення компонента.

Без цих важливих життєвих циклів, неможливо написати жодний веб-додаток.

2.1.4 Класи проти функцій

Існують два способи створення компонентів у React: компонент класу або функціональний компонент.

Компоненти — це основні будівельні блоки програми React. Вони дозволяють розбити інтерфейс користувача на фрагменти повторно використовуваних частин, які можна повторно використовувати і працювати незалежно один від одного [18].

Компонент класу, є звичайним класом, який розширює клас компонента бібліотеки React. Він називається компонентом з відстеження стану, тому що

він контролює зміну стану і реалізацію логіки компонента. Крім того, вони мають доступ до всіх фаз методу життєвого циклу React.

Функціональні компоненти – це просто функції JavaScript. До появи хуків у React 16.8 їх в основному називали компонентами без збереження стану або презентаційними компонентами, тому що вони тільки приймали і повертали дані для рендерингу в DOM.

Хуки дозволяють функціональним компонентам мати доступ до стану та інших функцій React. Тому компоненти класу, як правило, більше не потрібні.

З хуками React стає простішим. У React є два найчастіше використовувані хуки: хуки стану (useState) який відстежує стан додатку і хуки ефекту (useEffect) який дозволяє виконувати побічні ефекти у ваших компонентах.

Незважаючи на те, що функціональні компоненти виглядають простіше і їх швидше писати. Я віддаю перевагу класам, тому що для мене класові компоненти більш стабільні та типізовані. Оскільки я використовую React у зв'язці з мовою програмування TypeScript, який і був вигаданий як більш типізований аналог JavaScript, то мені більше підходять класові компоненти.

2.1.5 Розширення файлу JSX

JSX являє собою JavaScript XML. Файли JSX роблять додаток простим і зрозумілим. JSX полегшує звичайний JavaScript [13]. Він забезпечує можливість візуалізації компонентів з використанням HTML-подібного синтаксису. React використовує JSX для запису своїх компонентів.

JSX надає ряд елементів, призначених для відображення атрибутів, представлених у форматі HTML. Всі атрибути будуть отримані компонентом в якості реквізита.

2.1.6 Переваги та недоліки React

Переваги React:

- Легко зрозуміти та легко використовувати: Хороший запас документації, навчальних посібників та навчальних ресурсів допомагає розробникам легко вивчати та розуміти ReactJs.
- Просте створення динамічних веб-програм: На відміну від складного кодування динамічного веб-програми, React JS вимагає менше кодування і дає більше функціональності, а також підтримує створення машиночитаного коду.
- Багаторазові компоненти: Відомо, що програма ReactJS є багатокомпонентною програмою. Кожен компонент програми ReactJS можна використовувати повторно, і він відповідає за виведення невеликого повторно використовуваного фрагмента HTML коду. Будучи серцем всіх програм React, його компоненти також можуть бути вкладені один в одного.
- Висока продуктивність: ReactJS чудово справляється з керуванням віртуальним DOM. DOM має справу з HTML, XML або XHTML. Це кросплатформовий та програмний API.
- Підтримка зручних інструментів: Наявність зручного набору інструментів у ReactJS робить завдання розробників зрозумілим та легким.
- Наявність бібліотеки JavaScript: ReactJS пропонує багату бібліотеку JavaScript, що забезпечує гнучкість для розробників.
- Об'єм тестування кодів: За допомогою нативних інструментів ReactJS надзвичайно легко тестувати.

Недоліки React:

- Високий темп розвитку: Високі темпи розробки призводять до безперервних та швидких змін середовища настільки швидко, що важко прийняти все. Таким чином, навички розробників завжди повинні оновлюватись разом із змінами.

- Погана документація: При швидкому оновленні та швидкому прискоренні зробити правильну документацію непросто.
- JSX як бар'єр: JavaScript XML або JSX - це розширення синтаксису JavaScript, схоже на XML або HTML, використовуване ReactJS. Розроблений у дзвінки JavaScript React Framework, він розширює можливості ES6. Це дозволяє HTML-подібного тексту співіснувати з кодом реагування JavaScript.

2.1.7 SEO

SEO – це комплекс заходів для покращення сайту для його ранжування в пошукових системах [15].

За останні п'ять років Google оновив свою обробку контенту JavaScript, але сайти, які повністю відображаються на стороні клієнта, створюють інші проблеми, які необхідно враховувати. Важливо відзначити, що можна вирішити всі проблеми за допомогою React та SEO.

React JS – це інструмент розробки. React нічим не відрізняється від будь-якого іншого інструменту в стеку розробки, будь то плагін WordPress або обрана вами CDN. Від того, як він налаштований, буде залежить, чи погіршить він SEO чи покращить його.

В кінцевому рахунку, React хороший для SEO, оскільки він покращує користувальницький досвід.

SEO веб-сайтів на основі React має вирішальне значення. Нижче наведено кілька SEO-проблем:

2.2 Backend

Backend-розробка відноситься до розробки на стороні сервера. Основна увага приділяється базам даних, сценаріям, архітектурі веб-сайтів. Він містить закулісні дії, які відбуваються під час виконання будь-яких дій на веб-сайті.

Це може бути вхід в обліковий запис або здійснення покупки в інтернет-магазині. Код, написаний внутрішніми розробниками, допомагає браузерам взаємодіяти з інформацією бази даних.

В нашому випадку Backend буде складатися с наступного стеку технологій та засобів:

- .NET, C#
- Web API, Rest API
- MS SQL Express
- IIS Server.

2.2.1 .NET та мова програмування C#

.NET – це кросплатформна реалізація для запуску веб-сайтів, служб та консольних програм у Windows, Linux та macOS [25].

Без сумніву, .NET грає центральну роль індустрії розробки програмного забезпечення. Популярність .NET у спільноті розробників – це факт. Виміряти його можна за кількістю проектів з відкритим вихідним кодом по всьому світу та присутності C# у п'ятірці найпопулярніших мов програмування. Його популярність зростатиме ще більше, особливо після виходу останньої версії (.NET 5), яка зробила революцію у галузі, запустивши концепцію універсальної розробки програмного забезпечення.

C# (C-sharp) : сучасна об'єктно-орієнтована мова програмування, що належить до сімейства мов C. Його синтаксис може здатися знайомим розробникам C, C++, Java та JavaScript.

2.2.2 Web API, Rest API

Application Programming Interface (API) - інтерфейс з набором функцій, інструментів і протоколів для інтеграції прикладного програмного забезпечення та сервісів. Це дозволяє компаніям забезпечити безперешкодну взаємодію своїх продуктів та послуг з іншими продуктами та послугами, усуваючи необхідність нової інфраструктури підключення [21].

Оскільки, API допомагає взаємодіяти з різними програмами, організації можуть оптимізувати свої бізнес-операції для підвищення операційної ефективності.

Representational State Transfer (REST) є архітектурним стилем, що визначає набір обмежень для створення веб-сервісів. Попри поширену думку, це не інструмент, не протокол і не бібліотека. Натомість це веб-служба, яка взаємодіє між комп'ютерами або системами в Інтернеті.

Отже, REST API надає способом доступу до веб-сервісів без необхідності використання великих обчислювальних потужностей. REST API підтримуються архітектурою REST і також називаються RESTful API (див. рис 2.7).

Переваги REST API:

- Простота: оскільки REST API досить простий у освоєнні, можна організувати складні програми у прості ресурси. Він не тільки має можливість керувати високим навантаженням за допомогою проксі-сервера HTTP, але й дозволяє новим клієнтам працювати з потрібними програмами.
- Адаптивність: API REST допомагає легко передавати дані з одного сервера на інший. Він дає доступ зміни даних, що у базі даних, при розміщенні серверної і клієнтської частини різних серверах. Оскільки він певною мірою залежить від кодів, він допомагає синхронізувати дані на веб-сайтах без жодних проблем.

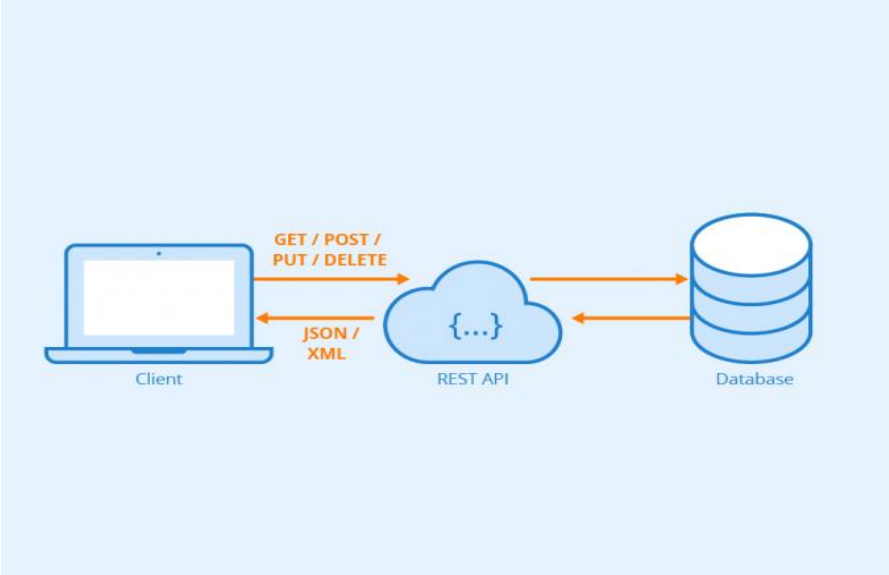


Рисунок 2.7 – Принцип REST API

Веб-API — це API, до якого можна отримати доступ через Інтернет за допомогою протоколу HTTP. Це платформа з відкритим вихідним кодом для веб-програми або веб-сервера, яка взаємодіє між клієнтами та веб-сайтами для доступу до даних. Оскільки Web API реалізує специфікацію протоколу, він включає такі поняття, як кешування, URI, управління версіями, заголовки і різні формати контенту. Розробники можуть створювати веб-API, використовуючи різні технології, такі як .NET, Java та багато інших. Веб-API дозволяє розробникам створювати служби, які можуть працювати з широким колом клієнтів, включаючи браузери, мобільні пристрої та планшети, як це показано на рисунку 2.8.

Переваги веб-API:

- Бізнес: оскільки веб-API є відкритим кодом, спрощується централізація логіки для підтримки узгодженості бізнес-даних. Оскільки дані з низькою пропускною здатністю (JSON/XML) легко аналізуються, вони легковажні і є ідеальним форматом обміну даними, їх можна інтегрувати з будь-якою мовою. Крім того, веб-API є невід'ємною частиною платформи ASP.NET, що робить його досить простим у обслуговуванні та розумінні.

- Технічні: одна з ключових технічних переваг веб-API полягає в тому, що вони не потребують складного налаштування. Будучи полегшеною архітектурою, вона ідеально підходить для пристроїв з обмеженою пропускною здатністю (смартфони). Вони підтримують OData (відкриті дані), маршрутизацію, прив'язку моделі та перевірки, що нагадують MVC.

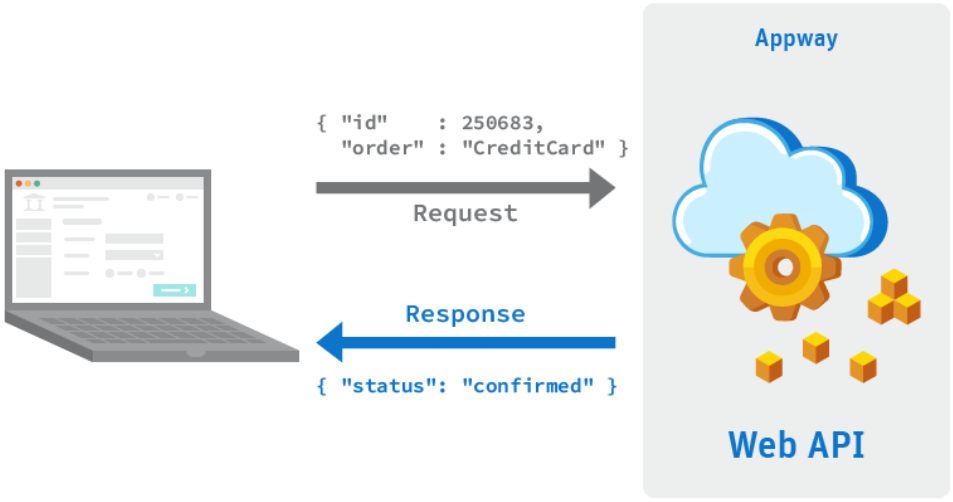


Рисунок 2.8 – Принцип Web-API

Веб-API можна розміщувати лише в інформаційній службі Інтернету (IIS) або самостійно, яка підтримує запити XML та JSON. Навпаки, REST API може розміщуватися лише у IIS, який підтримує стандартизовані запити XML.

Оскільки веб-API є спрощеною архітектурою, вони призначені для гаджетів, обмежених такими пристроями, як смартфони. Навпаки, API-інтерфейси REST відправляють та отримують дані по системах, що робить їх складною архітектурою.

Хоча API виконують ідентичні завдання, веб-API забезпечує гнучкість будь-якого стилю спілкування. У той час як API REST може використовуватися для зв'язку REST, SOAP і XML-RPC .

Веб-API підтримує протокол для протоколу HTTP/s та заголовки запитів/відповідей URL-адрес, які дозволяють службам досягати різних

клієнтів через Інтернет. З іншого боку, весь зв'язок у REST API підтримується лише через протокол HTTP.

2.2.3 MS SQL Express

SQL Server Express — це повнофункціональна база даних, яку можна розгорнути на сервері або вбудувати у програму. Express безкоштовний і має багато функцій корпоративної версії. SQL Server Express, ймовірно, найбільше підходить для підтримки виробничих програм для малих і середніх підприємств. Типовим варіантом використання SQL Server Express буде розгортання розробниками, які не хочуть створювати програми з базою даних, розміщеною на сервері. Використовуючи Express, можна розробляти програми зі своєю базою даних SQL Server [26].

Переваги SQL Server Express:

- Однією з величезних переваг SQL Server Express є те, що він є безкоштовним.
- SQL Server Express є ідеальною відправною точкою для невеликих незалежних постачальників програмного забезпечення (ISV), оскільки його можна використовувати з будь-якою невеликою програмою.
- У SQL Server Express є можливість безкоштовного онлайн резервного копіювання, яке допоможе захистити ваші цінні бізнес-дані, якщо щось піде не так.
- Хоча Express є «полегшеною» версією SQL Server, в ній все ж таки є вражаючий набір функцій, за які вам доведеться платити в інших системах.

2.2.4 IIS Server

Веб-сервер IIS працює на платформі Microsoft .NET в Windows. Хоча можна запустити IIS на Linux та Mac за допомогою Mono, це не

рекомендується і, швидше за все, буде нестабільним [24]. Він універсальний і стабільний і вже багато років широко використовується у виробництві (див. рис 2.9).

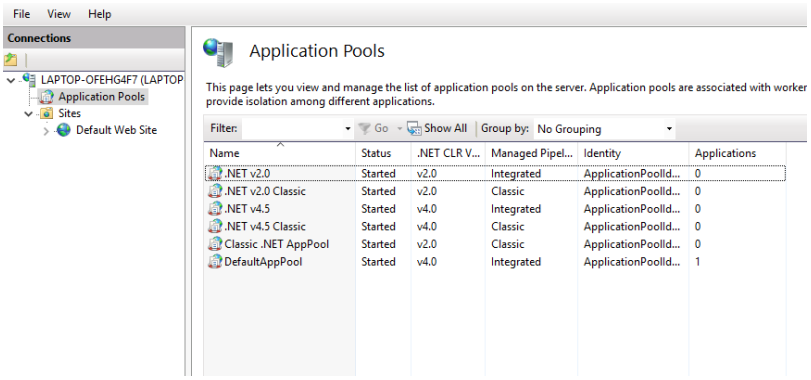


Рисунок 2.9 – Зовнішній вигляд інтерфейсу IIS Server

IIS багатий на функції. Найчастіше IIS використовується для розміщення веб-застосунків ASP.NET і статичних веб-сайтів. Його також можна використовувати як FTP-сервер, розміщувати служби WCF і розширювати для розміщення веб-програм, створених на інших платформах, таких як PHP.

2.3 Мова програмування TypeScript та її сфери використання

TypeScript — мова програмування, представлена Microsoft в 2012 році і позиціонується як засіб розробки веб-додатків, що розширює можливості JavaScript [20].

TypeScript є сумісним з JavaScript і компілюється в останній. Фактично, після компіляції, програму на TypeScript можна виконувати у будь-якому сучасному браузері або використовувати разом із серверною платформою Node.js. Під ліцензією Apache поширюється код експериментального компілятора, що транслює TypeScript в JavaScript. Його технологія ведеться в громадському репозиторії через обслуговування GitHub.

Синтаксично, TypeScript дуже схожий на JScript .NET, чергову реалізацію Microsoft мовного стандарту ECMA-262, що забезпечує підтримку статичної типізації та класичних об'єктно-орієнтованих можливостей мови, таких як класи, спадкування, інтерфейси та простори імен.

З TypeScript можна використовувати існуючий JavaScript код, включати популярні бібліотеки JavaScript, і викликати TypeScript код, згенерований з інших JavaScript. Оголошення типів цих бібліотек поставляються разом з вихідним кодом.

TypeScript відрізняється від JavaScript можливістю явного статичного призначення типів, підтримкою використання повноцінних класів (як у традиційних об'єктно-орієнтованих мовах), а також підтримкою підключення модулів, що покликане підвищити швидкість розробки, полегшити читання, рефакторинг та повторне використання коду, допомогти здійснювати пошук помилок на етапі розробки та компіляції, і, можливо, прискорити виконання програм [12].

Переваги TypeScript:

- Сильна статична типізація. JavaScript не є строго типізованим. TypeScript поставляється з додатковою статичною типізацією та системою виведення типів через TLS (мовна служба TypeScript). Тип змінної, оголошеної без зазначення типу, можна визначити TLS з урахуванням її значення.
- TypeScript підтримує визначення типів для наявних бібліотек JavaScript. Файл визначення TypeScript містить визначення для зовнішніх бібліотек JavaScript.
- TypeScript підтримує концепції об'єктно-орієнтованого програмування, такі як класи, інтерфейси, успадкування тощо.

В основі TypeScript лежать такі три компоненти, які зображено на рисунку 2.10:

I. Мова - включає синтаксис, ключові слова та анотації типів.

II. Компілятор TypeScript — компілятор TypeScript (tsc) перетворює інструкції, написані на TypeScript, на їх еквівалент JavaScript.

III. Мовна служба TypeScript — «Мовна служба» надає додатковий рівень навколо основного конвеєра компілятора, який є програмою, подібною до редактора. Мовний сервіс підтримує стандартний набір типових операцій редактора, таких як завершення операторів, довідка з підпису, форматування та виділення коду, розфарбовування тощо.

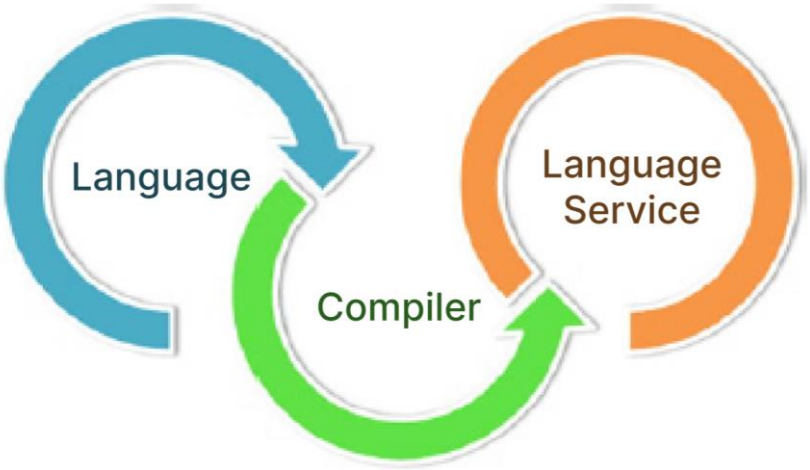


Рисунок 2.10 – Основа мови програмування TypeScript

Що стосується сфери використвння, то у Frontend розробці можна виділити лише дві мови, JavaScript та TypeScript. JavaScript більш популярний, але щороку все більше програмістів віддають перевагу TypeScript (див. рис 2.11).

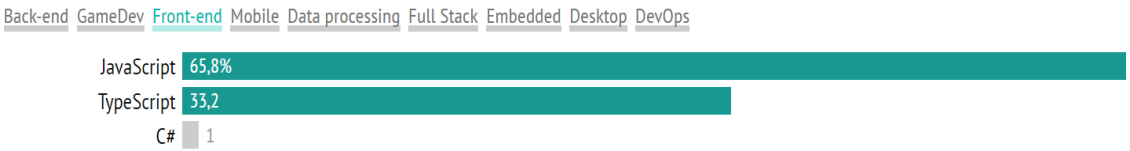


Рисунок 2.11 – Мови програмування за сферами використання

А на рисунку 2.12 ми можемо побачити, що рейтинг індексу вподобання, TypeScript стрімко зростає, це не скажеш про JavaScript [6].

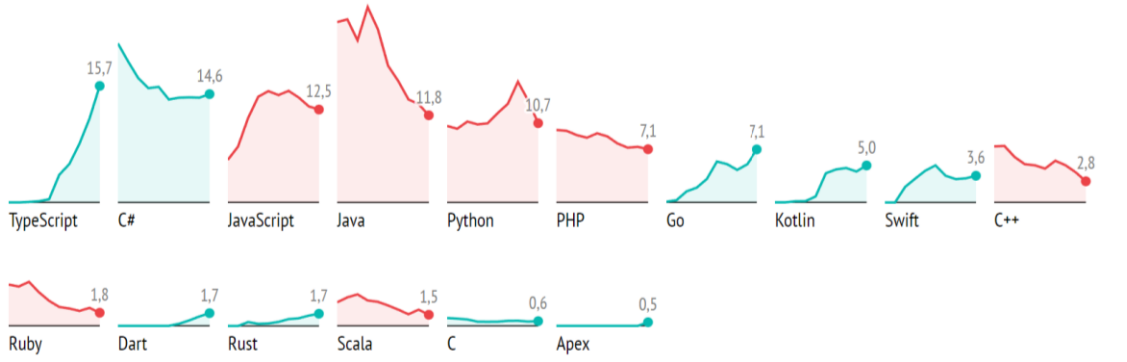


Рисунок 2.12 – Індекси вподобання різних мов програмування

Можна сказати, якщо досвідчений розробник працює над відносно невеликими проектами, то JavaScript ідеальний. Однак, якщо є команда розробників зі знаннями та досвідом, то Typescript є найкращим варіантом.

2.4 React Syncfusion

Бібліотека компонентів інтерфейсу користувача Syncfusion React — це єдиний набір, який вам коли-небудь знадобиться для створення програми, оскільки він містить понад 65 високопродуктивних, легких, модульних і адаптивних компонентів інтерфейсу користувача в одному пакеті.

Нижче, на рисунку 2.13, показано, огляд Syncfusion

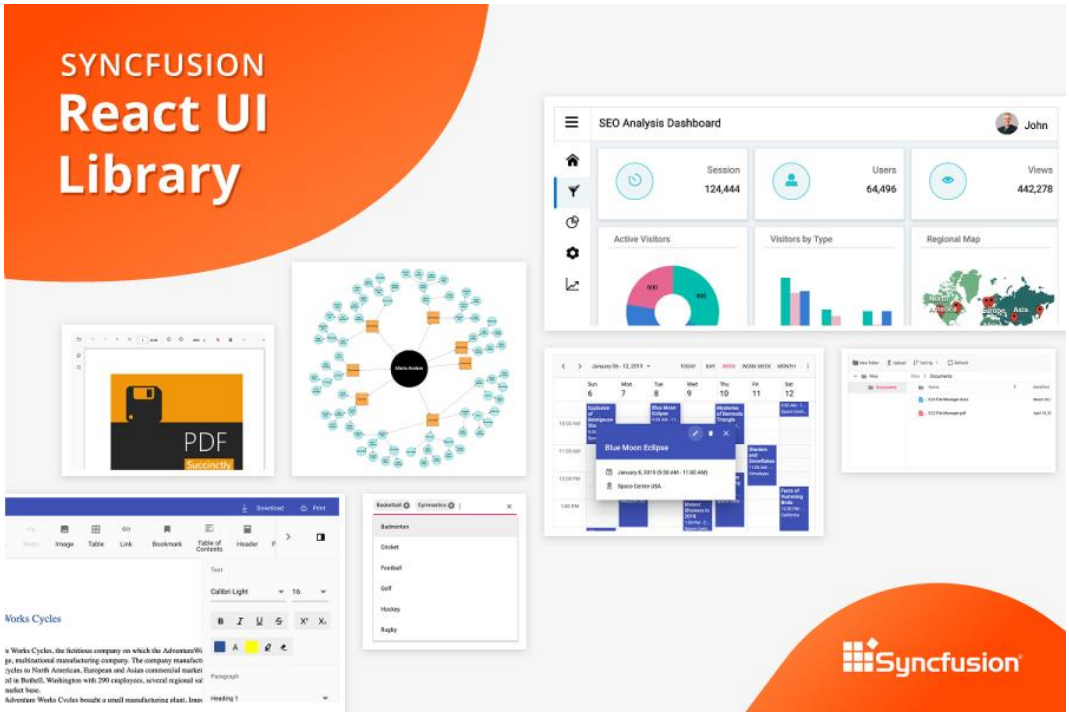


Рисунок 2.13 – Огляд Syncfusion

Причини використання React Syncfusion:

- Всі компоненти були побудовані як модулі, щоб забезпечити вибіркове звернення, тому додаток включені ті компоненти і функції, які потрібні.
- Вся бібліотека створена з нуля, щоб бути легкою та модульною. Його площа можна зменшити, включивши лише певні компоненти та функції, які потрібні програмі.
- Усі компоненти підтримують сенсорне керування та відображаються адаптивно залежно від пристрою, забезпечуючи оптимальну взаємодію з користувачем на телефонах, планшетах та настільних комп'ютерах.
- Повний вихідний код, файли модульних тестів та сценарії тестування доступні на GitHub.

З існуючих аналогів, React Syncfusion є одним з кращих рішень.

2.5 Середа розробки WebStorm

WebStorm — інтегрована середовище розробки для JavaScript, а також для CSS та HTML, що розроблене на основі платформи IntelliJ від компанії JetBrains. Як і інші IDE JetBrains, WebStorm дозволяє автоматизувати рутинну роботу та легко справлятися зі складними завданнями, роблячи розрабку більш цікавою [22].

Середа розробки WebStorm, має такий вигляд рисунок 2.14.

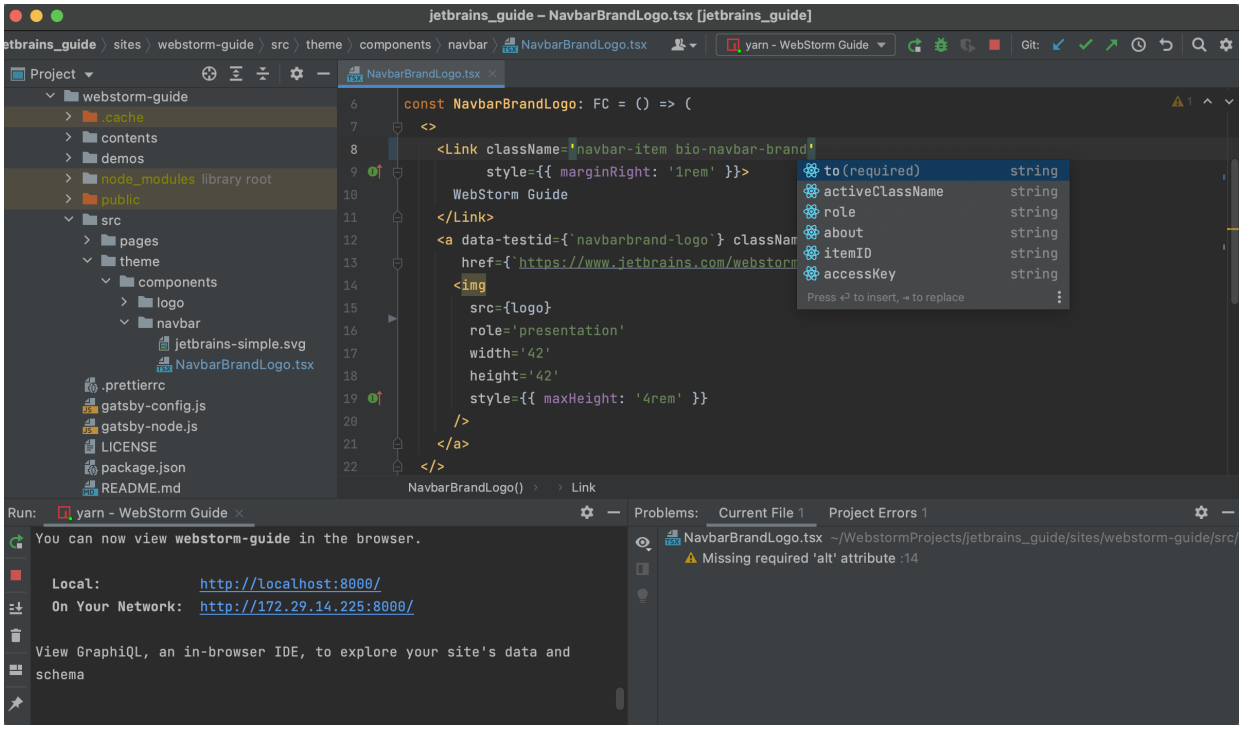


Рисунок 2.14 – Вигляд середи розробки WebStorm

WebStorm забезпечує налагодження JavaScript і надає широкий діапазон можливостей: знаходження точки зупинки в HTML і JavaScript, налаштування параметрів точки зупинки, тестування синтаксису коду в режимі реального часу. Продукт підтримує платформи JQuery, Prototype, DoJo, MooTools, Qooxdoo та Bindows. WebStorm запобігає інтегровану перевірку тексту на теги, помилки в написанні та послідовність коду [11]. WebStorm дозволяє

редагувати файли та автоматично синхронізувати їх на вимогу при віддаленій роботі або зберіганні.

2.6 Висновки за розділом

У цьому розділі було виявлено що розробка web додатку має проводиться на мові TypeScript та з використанням бібліотеки React. За елемент управління діаграм відповідає React Syncfusion. За середовище розробки було обрано найпопулярнішу, у цій сфері, середовище програмування – WebStorm. Було зроблено аналіз та порівняння технологій для написання веб-застосунку. В ході аналізу, було з’ясовано переваги та недоліки вище перелічених технологій та засобів.

РОЗДІЛ 3
ПРОГРАМНИЙ МОДУЛЬ

3.1 Проектування системи

Замовником було поставлено завдання створити front-end частину веб-сайту на базі single-page application (SPA). Також було надано список технологій, за допомогою яких і необхідно створити веб-сайт:

- React
- TypeScript
- Syncfusion React

Веб-застосунок має виконувати функції планування дій при міграції великих обсягів даних усередині компанії замовника. Він повинен допомагати користувачеві не заплутатися у цьому нелегкому процесі. Саме планування має відбуватися у вигляді діаграм (UML) і таблиць, які пов'язані між собою. Тобто компанія хоче отримати внутрішній продукт, для особистого використання. Це дуже унікальне рішення, і в цьому його складність.

Веб-додаток складається з двох частин:

1. Backend – В нашому випадку backend частина проекту, вже готова. Ця частина складається за наступних технологій: .NET C#, MS SQL Express, Web API, Rest API, IIS.
2. Frontend – SPA, React, TypeScript, Syncfusion.

На рисунку 3.1 зображено діаграму прецедентів вкладки процесів веб-застосунку. Це може підмітити на ті дії, які може робити актор (користувач).

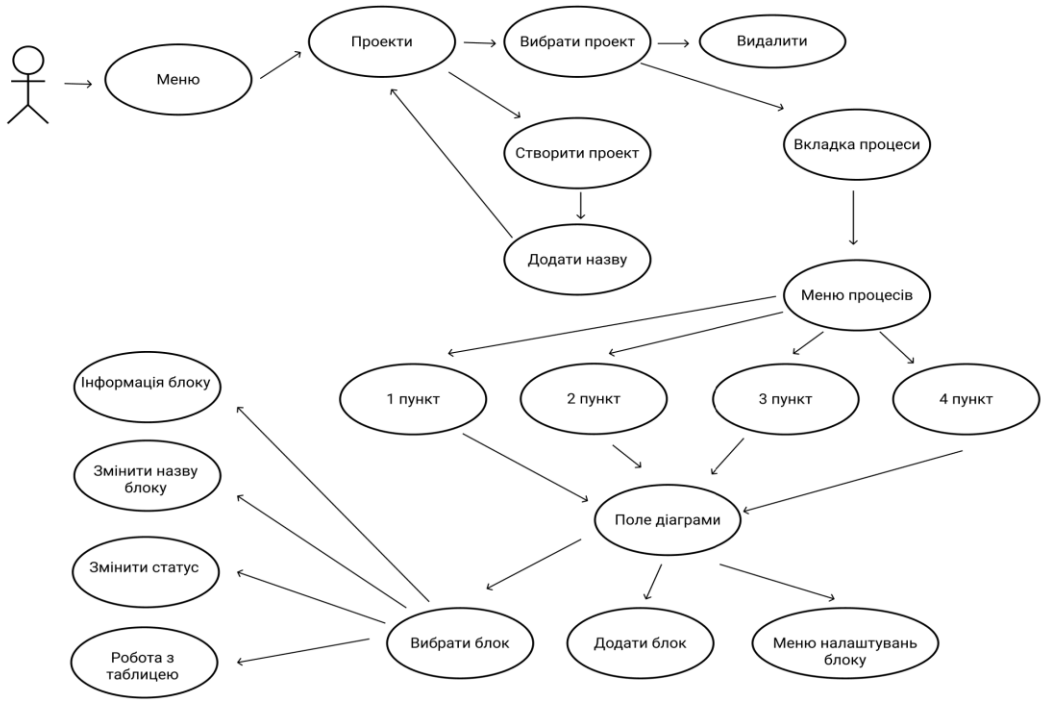


Рисунок 3.1 – Діаграма прецедентів вкладки процесів

Діаграма містить одного актора – це користувач.

Користувач при попаданні на головну сторінку може відкрити головне меню та вибрати проект, або якщо проектів немає, то він може його створити натиснув на кнопку “Add” та додавши назву проекта. Після того як користувач обрав проект, далі він може перейти в вкладку процесів та обрати один з чотирьох пунктів меню. Кожний пункт містить один шаблон. Після вибору вкладки меню, можна працювати з діаграмою, а саме додавати нові блоки, з'єднання, статус, додавати ім'я блоку, вибирати блок, видалити блок, тощо. Після того як користувач обрав блок, він може редагувати назву блоку, додавати інформацію, та працювати з таблицею.

Далі необхідно описати яка діаграма прецедентів буде відноситися до інших вкладок проекту. Вище я описав дії користувача на вкладці процесів, але в нас є також інші вкладки. На рисунку 3.2 зображено діаграму прецедентів інших вкладок.

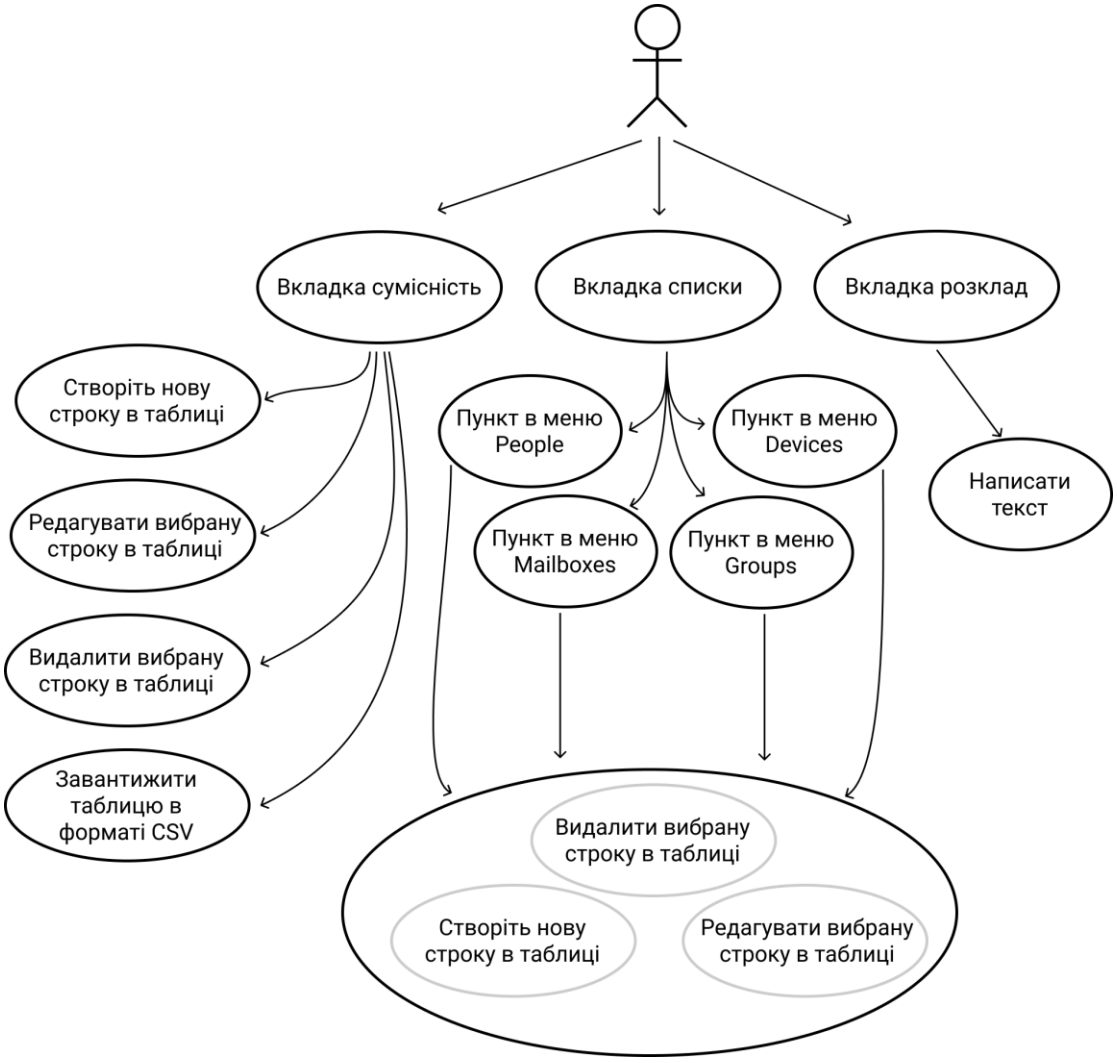


Рисунок 3.2 – Діаграма прецедентів інших вкладок

Користувач бачить три вкладки: сумісність, списки, розклад.

Вкладка сумісність є таблицею з сорок однією колонкою. У цій таблиці ми можемо додавати, редагувати та видаляти рядки. Також ми можемо зберегти таблицю як файл. Дані цієї таблиці автоматично записуються на локальний сервер сайту. Насправді ця таблиця несе лише інформативний характер.

Наступна вкладка – це списки. Тут користувач може відкрити в боковому меню, одну з чотирьох вкладок, в якій знаходиться таблиця. У нас є основна вкладка – це People. У цій вкладці є таблиця з можливістю розгорнути контент будь-якого рядка, це називається row expander. У контенті рядка,

знаходяться три таблиці, це ті три таблиці які знаходяться в вкладках меню, що залишилися. Ми можемо додавати рядок таблиці People, рядки інших таблиць за ключами.

Остання вкладка – це розклад. Тут ми можемо написати простий текст. Всі дані у всіх вкладках зберігаються на локальний сервер.

3.2 Архітектура додатку

Web додаток складається з компонентів:

- Головний компонент
- Меню
- Панель вкладок
- Процеси
- Сумісність
- Списки
- Розклад
- Діаграми
- Меню вкладки процесів
- Бокова інформація
- Таблиці
- Модальні вікна
- Текстові форми
- Фільтр
- Інформаційні блоки
- Файловий менеджер

Для реалізацій усього вище переліченого, крім діаграм та таблиць було розроблено власне рішення. Перебуваючи під підпискою про нерозголошення я не можу показувати всі деталі коду продукту та описувати класову та файлову архітектуру крім діаграм та таблиць, бо вони не є власним рішенням.

3.2.1 Головний компонент

Все починається з головного компонета. До головного компонету додається панель вкладок, меню та пусте поле, якщо проект не обраний. Меню видає список існуючих проектів і надає можливість менеджменту проектів, тобто додати та видалити (рис. 3.3). Усі проекти зберігаються на локальному сервері.

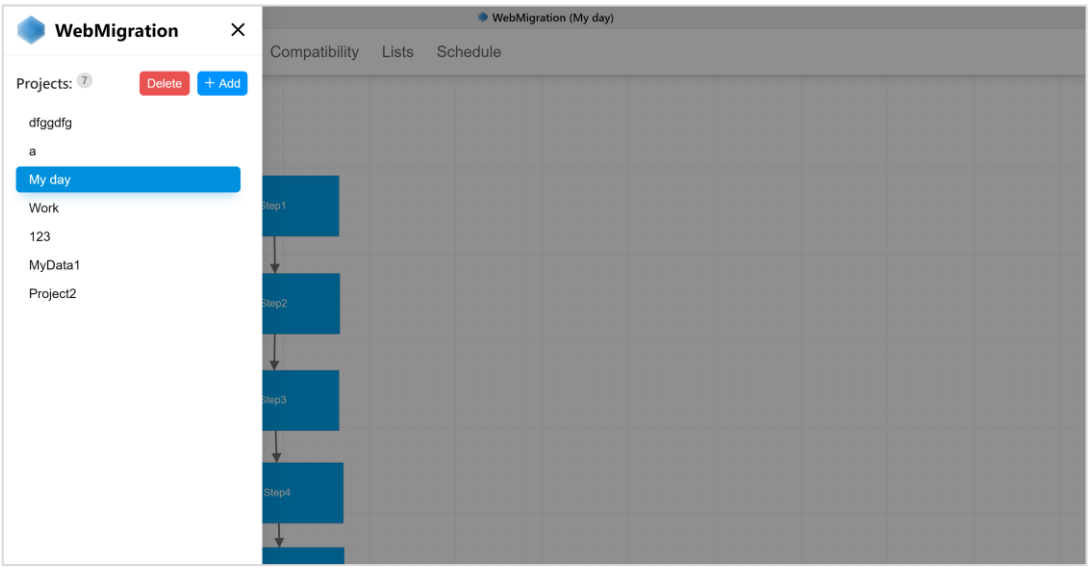
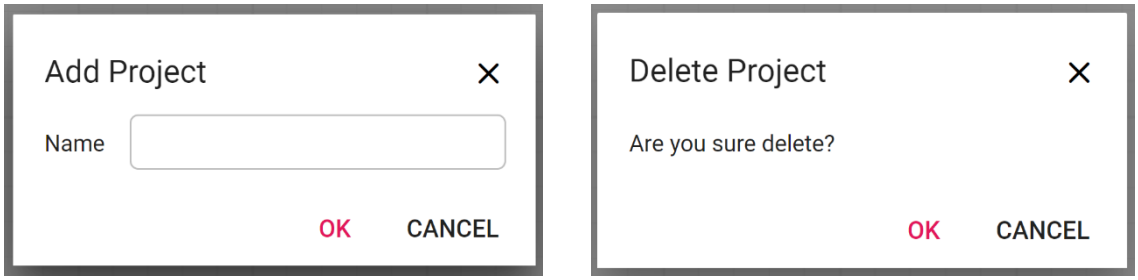


Рисунок 3.3 – Меню проектів

Менеджмент проектів, а це видалення та створення відбувається за допомогою модальних вікон (рис 3.4).



a) b)

Рисунок 3.4 – Видалення та створення проектів.

Модельні вікна були створені за допомогою Syncfusion, це можна побачити на лістінгу 3.1

Лістинг 3.1 – Створення модельних вікон

```
<div  className="d4"><Menu selectionItem={selectionItem} onSelectedItem={setSelectionItem}
onAdd={()=>{
                    refDialogComponent.current?.show()
                  }} onDelete={()=>{
                    refDialogComponent2.current?.show();
                  }}  items={dataProjects} active={menuActive}
setActive={setMenuActive}></Menu></div>
</div>
```

Для інформативності, необхідно додати шапку для відображення інформації про проект який відкритий.

Після відкриття проекту, необхідно зробити запит до серверу задля отримання даних про проект, та подальшого відображення.

Контент вкладки процесів з'являється після того як користувач обрав проект (рис. 3.5). В контенті необхідно додати меню з 4 пунктів та компонент діаграми. Компонент діаграми являє собою продуктом компанії Syncfusion і має назву “Diagram”. Для його додаванню до проекту, необхідно установити пакети Syncfusion в cmd проекту: “npm install @syncfusion/ej2-react-diagrams –save”. Після успішного завершення установки, ми можемо додати діаграму до проекту, це можна побачити на лістингу 3.2.

На рисунку 3.5, ми також можемо побачити в нижньому лівому куту, кнопку синього кольору з зображенням плюсу. Ця кнопка відповідає за додавання нових блоків діаграми. Коли ми натискаємо на цю кнопку, в кінець діаграми додається блок з назвою Step та номером блоку.

Лістинг 3.2 – Додавання діаграми до проекту

```
import * as React from "react";
```

```
import "./App.css";
import { DiagramComponent } from "@syncfusion/ej2-react-diagrams";
export default class App extends React.Component<{}, {}> {
  render() {
return <DiagramComponent id="diagram" />;
  }
}
```

Усі блоки мають можливість перейменування на подвійний клік по блоку. Але не тільки блоки мають можливість приймати назву. Таку можливість мають також стрілочки.

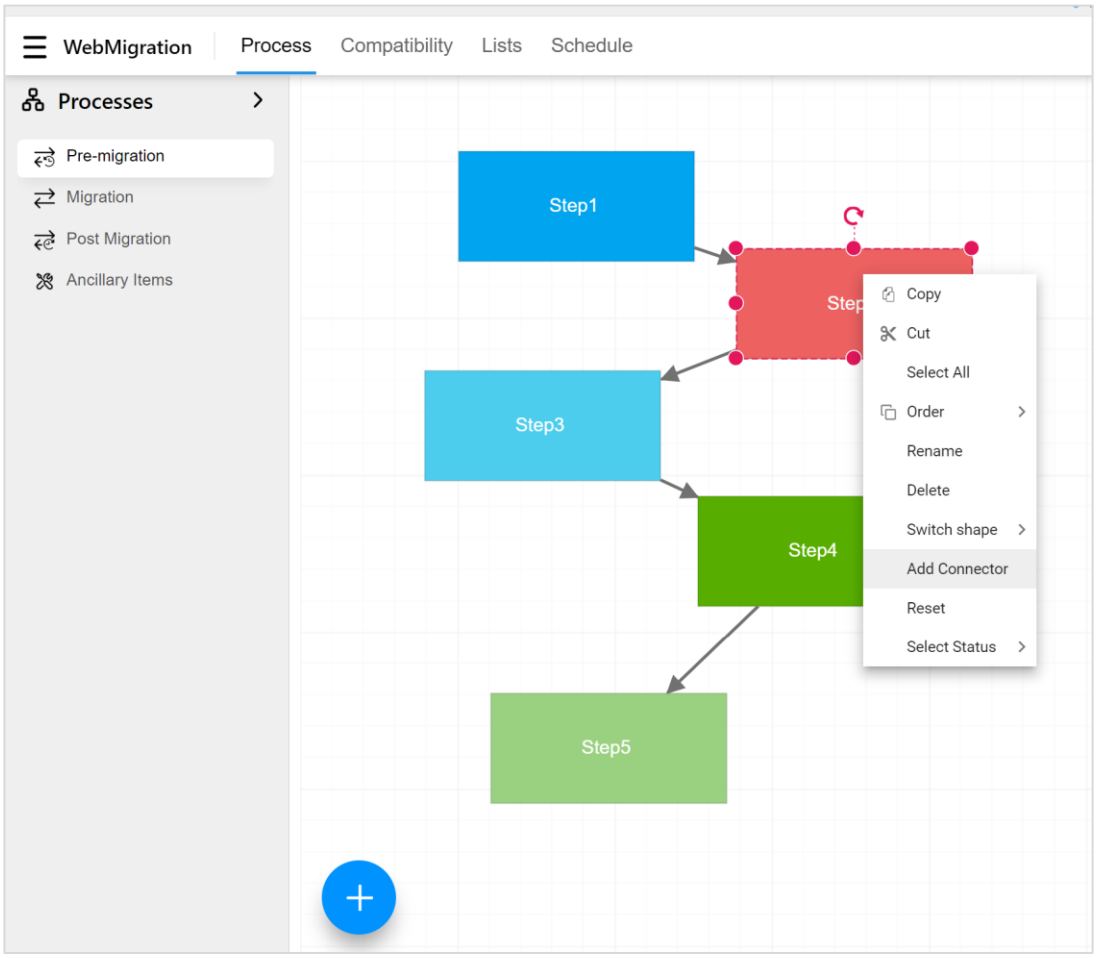


Рисунок 3.5 – Діаграма

Також ми маємо контекстне меню, яке теж можна побачити на зображенні 3.5. Це меню відповідає за більшість можливостей роботи з блоками, це дуже зручно. Меню має наступні пункти:

- Копіювання виділеної області в буфер обміну.
- Вирізання та подальше копіювання в буфер обміну виділеної області.
- Вибрати все. Тобто ми робимо виділеним все що є на сітці.
- Перейменування того що було виділено.
- Видалення виділеної області.
- Змінити фігуру на логічний оператор або звичайний блок.
- Додати до блоку стрілочку, або іншими словами вказівник.
- Очистити всю сітку.
- Вибрати статус, від якого залежить колір та назва статусу блоку.

Якщо ми визиваємо контекстне меню по пустому полю то в нас з’являються три нових пункти:

- Вставити те що знаходиться в буфері обміну, проекту.
- Відмінити змини.
- Скасувати відміну змін.

Буфер обміну має властивість очищатися якщо ми змінюємо вкладку або проект.

Ми можемо змінювати тип фігури з звичайного блоку на логічний блок, де є два вказівника, так або ні. Вибір типу фігури, можна зробити через контекстне меню. На зображенні 3.6 ми можемо помітити як працює логічний блок.

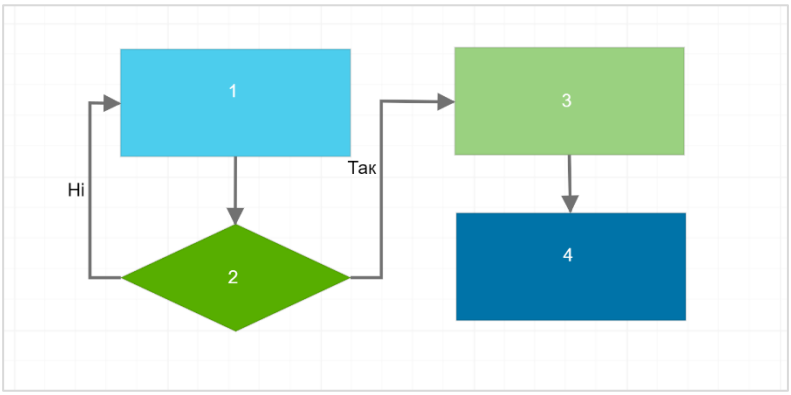


Рисунок 3.6 – Логічний блок

Далі необхідно додати бокову інформацію, яка буде з'являтися після того як користувач обрав два ризи натисне на блок (рис. 3.7). На цій формі знаходиться хедер і контент. На хедері ми можемо помітити кнопку закриття форми, назва, статус та кнопка видалення блоку. Якщо натиснути на видалення то з’явиться вікно с підтвердженням намірів.

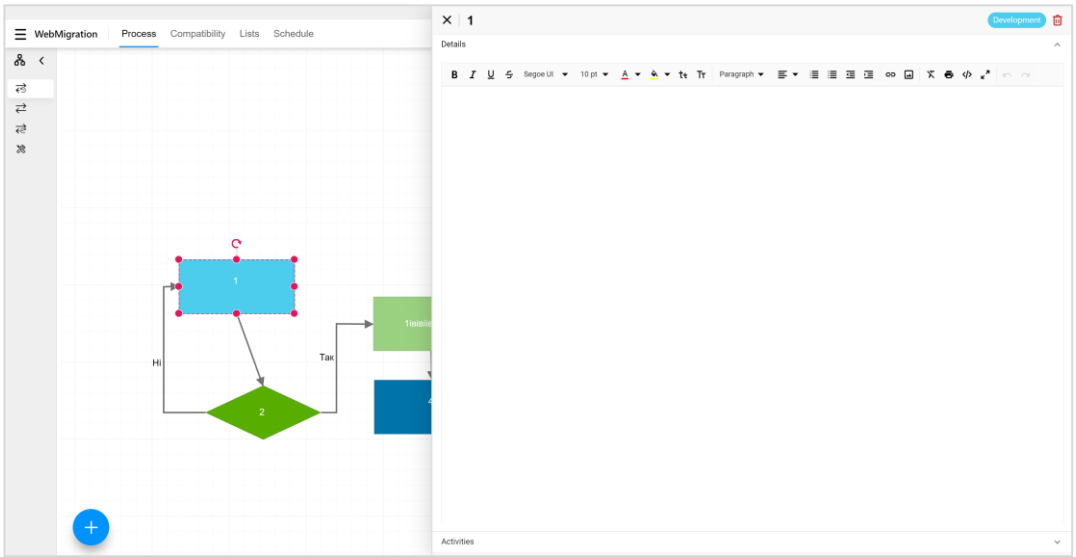


Рисунок 3.7 – Бокова інформація

Щодо контенту, то там є компонент expander, в якому є два поля: деталі та активності.

В формі деталей, знаходиться дуже цікавий компонент, який схожий на Microsoft Word (рис 3.8). За допомогою цього компонету ми можемо надавати тексту різних кольорів, шрифтів, розмірів, стилів. Він нагадує класичний Microsoft Word. Є навіть можливість розгорнути на весь екран, зробити печать, зробити посилання та вставити зображення. Це має бути зручно, ми можемо в деталі записати данні в зручній нам формі.

Кожен блок може бути унікальним за контентом, бо ми не знаємо який блок за що буде відповідати. Це вирішує для себе користувач.

Робити зручні посилання на статті або зображання задля спрощення прощення процесу міграції. Виділяти жовтим маркером шо важливо. Це і не тільки, робить застосунок унікальним.

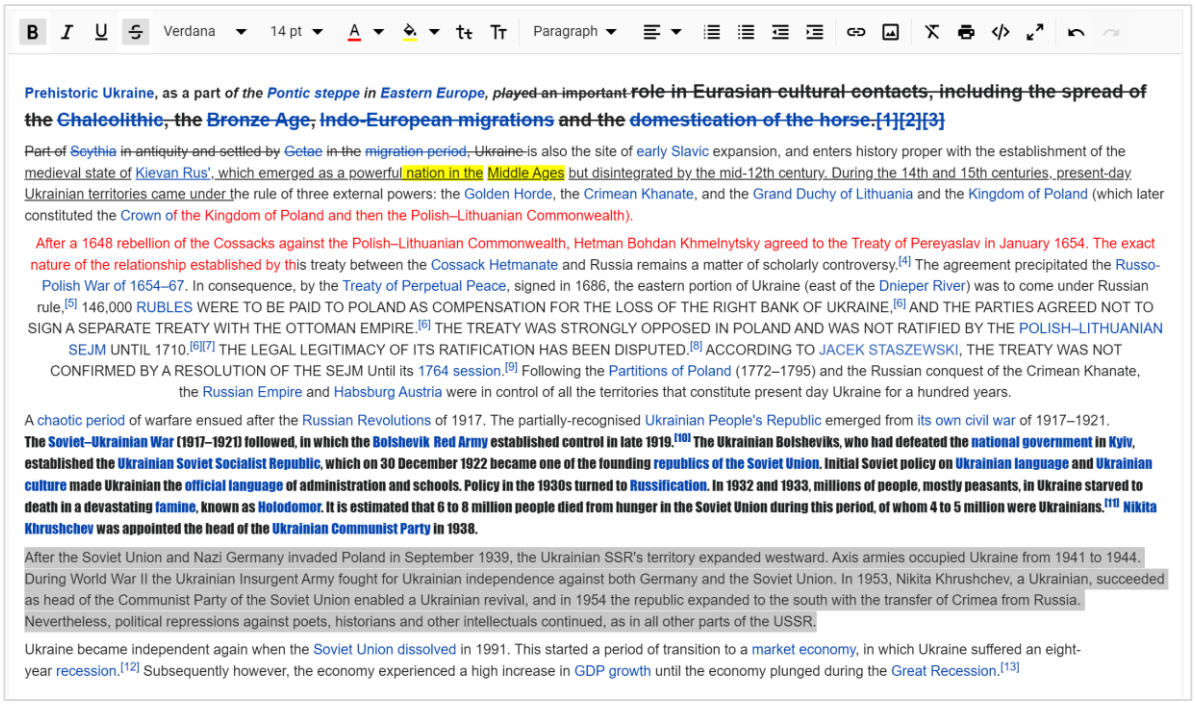


Рисунок 3.8 – Компонент для наповнення сторінки інформацією

Наступний компонент відповідає за відображення таблиці (рис. 3.9). Ця форма називається активності. В цій таблиці відображаються усі активності по блоку, на якому вони етапі, який мають статус, тощо. Ми можемо як додавати нові рядки в таблицю, так і редагувати та видаляти.

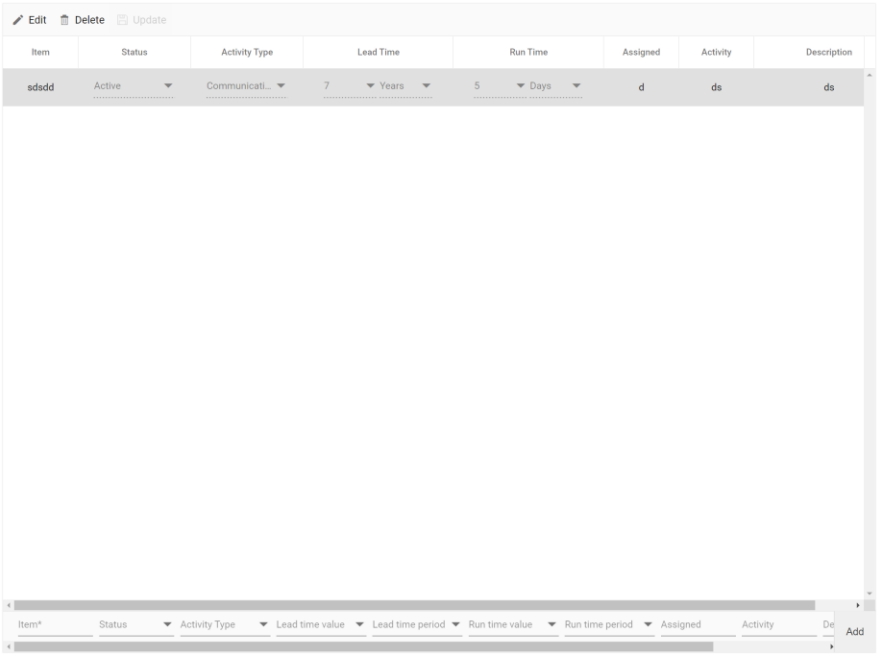


Рисунок 3.9 – Вигляд таблиці активності

Додавання нової сторінки відбувається трохи не звичайним чином. Потрібно заповнити усі поля які унизу, та натиснути на кнопку додати.

Подалі такий спосіб додавання сторінок не буде використовуватися. Додавання буде відбуватися за допомогою модального вікна.

Для створення таблиці ми додаємо до проекту пакет: “npm install @syncfusion/ej2-react-grids --save ” і додаємо компонет “Grid”.

Додавання пакетів в проект, робиться через консоль проекту, куди и треба писати команди.

Після того як було додано до проекту компонент, необхідно створити саму таблицю як показано на лістінгу 3.3

Лістинг 3.3 – Додавання Grid до проекту

```
return <GridComponent dataSource={data}>
        <ColumnsDirective>
          <ColumnDirective field='OrderID' width='100' textAlign="Right"/>
          <ColumnDirective field='EmployeeID' width='100' textAlign="Right"/>
          <ColumnDirective field='Freight' width='100' format="C2" textAlign="Right"/>
          <ColumnDirective field='ShipCountry' width='100'/>
        </ColumnsDirective>
```

```
</GridComponent>
```

Щодо хедера таблиці, то там розташовані кнопки редагування, видалення та оновлення.

Також, для редагування можна просто два рази натиснути на строку в таблиці.

3.2.2 Вкладка сумісність

В контенті вкладки є тільки таблиця яка складається з сорок однієї колонки. У цій таблиці ми можемо додавати, редагувати та видаляти рядки. Також ми можемо зберегти таблицю як файл. Дані цієї таблиці автоматично записуються на локальний сервер сайту. Усе вище перелічене можна помітити на зображенні 3.10.

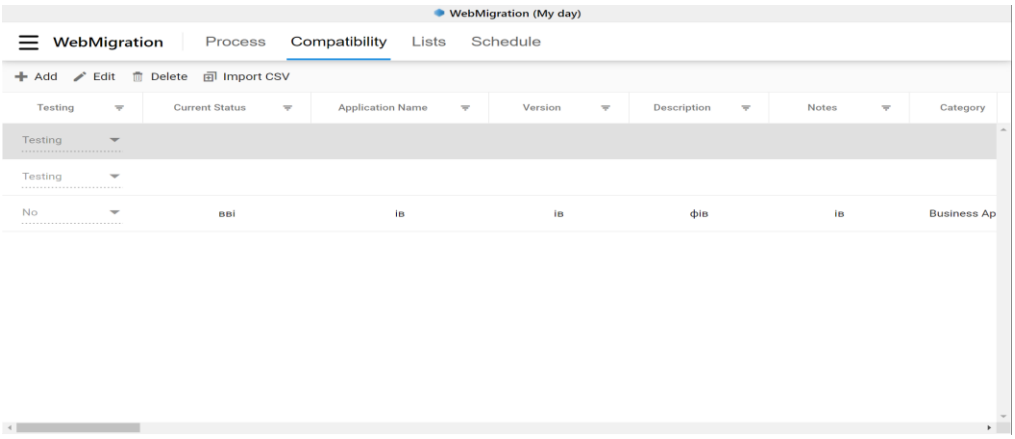


Рисунок 3.10 – Таблиця сумісності

Усі дії виконуються через модальне віко. Це дуже зручно та естетично бо у нас є дуже багато полів в які треба вводити данні (рис. 3.11).

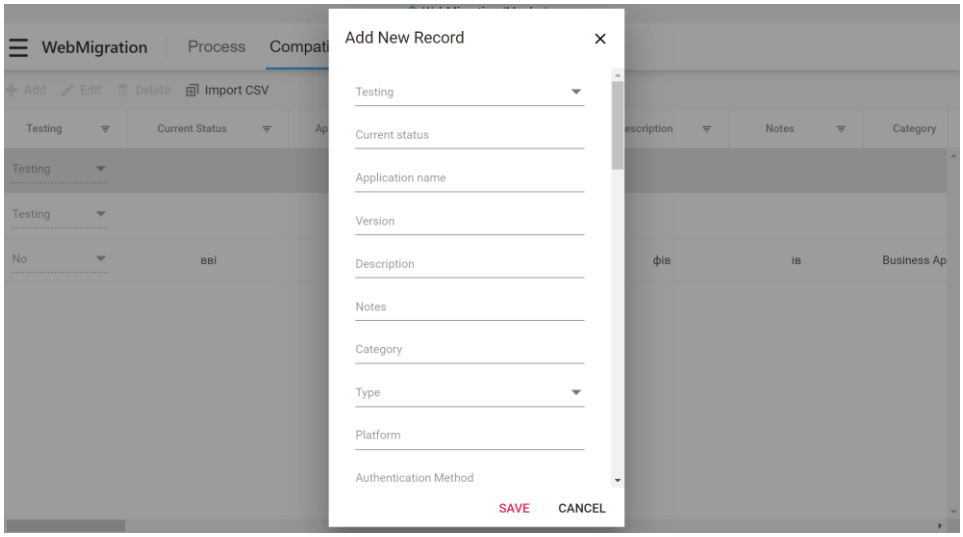


Рисунок 3.11 – Модальне вікно на додавання

В цій таблиці є можливість імпортувати данні в форматі файлу CSV. Це означає що можна відкрити, файл CSV и далі працювати з цими даними. Така функція має бути зручною.

Лістинг 3.4 – Додавання імпорту CSV до таблиці сумісності

```
<DialogComponent width='250px' buttons={buttons} content={'Malformed data, file import failed. Error: '.concat(errorMsg)} header='Error' visible={showError}/>
<GridComponent toolbarClick={tbClick} actionComplete={actionComplete.bind(this)} ref={gridComponentRef} allowFiltering={true} filterSettings={filterOptions} dataSource={gridDataSource} allowResizing={true} height={'calc(100vh - 170px)'} width='auto' selectionSettings={{type: 'Single' }} editSettings={{allowEditing: true, allowAdding: true, allowDeleting: true, showDeleteConfirmDialog: true, mode: 'Dialog' }} toolbar={['Add','Edit', 'Delete', { text: "Import CSV", tooltipText: "Upload CSV file", prefixIcon: "e-expand", id: "Click" } ]} />
```

Таблиця має тільки інформаційний характер, вона не впливає на інші таблиці. Нам необхідно зберігати данні про данні проектів, а також ті данні які знаходяться всередині проектів.

Зберігання буде відбуватися на локальному сервері написаному на C#, через RestAPI. А допоможе в цьому React Feath

Лістинг 3.5 – Функція запиту до серверу

```
async function fetcher(url: RequestInfo) {
    try {
      const res = await fetch(url, {mode: 'cors'});
      const json = await res.json();
      setIsLoading(false);
      return json;
    } catch (e) {
      throw e;
    }
  }
```

Тобто при необхідності ми можемо як зберігати так загрузити данні о проектах та що знаходяться всередині проектів.

У кожної колонки присутній фільтр. Він відображається за допомогою модального вікна. Те як виглядає фільтр можна помітити на зображанні 3.12.

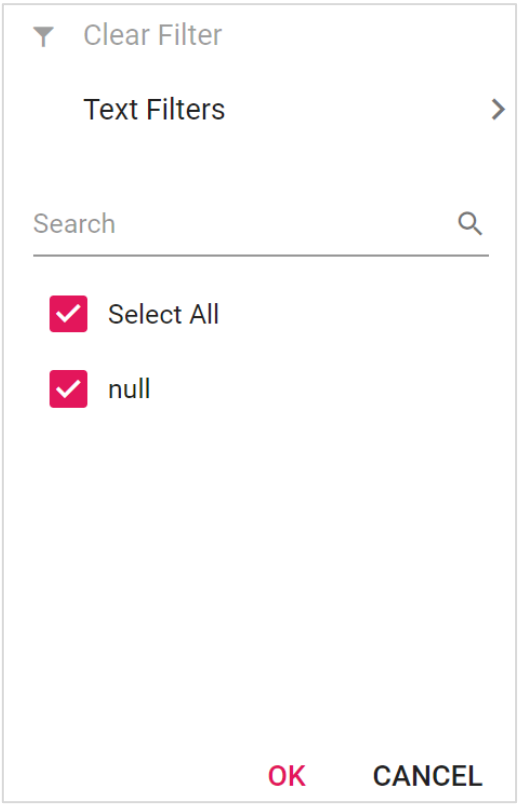


Рисунок 3.12 – Фільтр колонки

Для того щоб додати можливість фільтру по колонці, нам необхідно вказати налаштування колонок таблиці, фрагмент коду можна помітити на лістингу 3.6.

Лістинг 3.6 – Надання можливості фільтру по колонкам.

```
let  filterOptions: FilterSettingsModel = {
   type: 'Excel'
};
```

Тут ми встановили параметр за яким буде проходити фільтрування. Таким чином ми маємо гарно працюючий фільтр для зручності пошуку даних. Фільтр повністю заміняє нам класичний пошук.

3.2.3 Вкладка списки

Вкладка списків представляє собою табір таблиць які пов’язані між собою. На зображанні 3.13 можна побачити бокове меню з чотирма пунктами: people, mailboxes, devices та groups. Контент цих вкладок представляє собою таблиці, які пов’язані між собою.

Ми маємо стандартний набір можливостей для взаємодії з таблицею. Можна додати новий рядок, видалити вибрану область, редагувати обраний рядок та оновити таблицю. За ці дії відповідають кнопки, які знаходяться у хедері таблиці.

В вкладці people, знаходиться основна таблиця, в якій рядки мають можливість розгортатися. В розгорнутому виді, з’являється три додаткові таблиці: devices, mailboxes та groups.

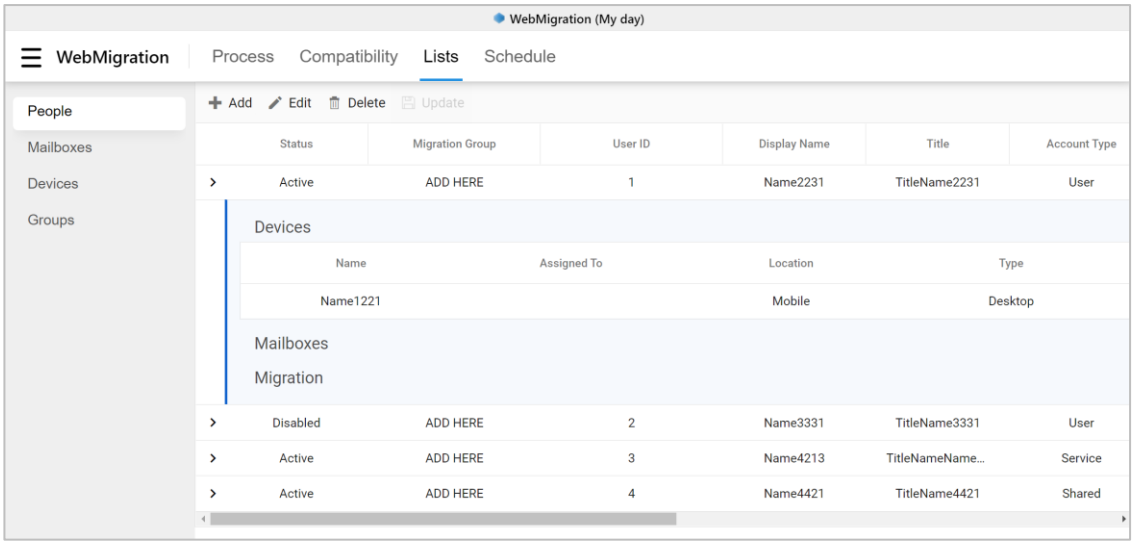


Рисунок 3.13 – Вкладка списків

В таблиці people, у кожного рядка, є унікальне поле, UserID. При створенні нового поля, UserID повинен бути унікальним, тобто не співпадати з іншими UserID в таблиці.

Коли я вношу якісь зміни в таблицю, то в таблиці визивається подія actionComplete. Тут я перевіряю, якщо подія на збереження даних, то я також перевіряю, чи співпадає поле UserID з іншими. Якщо так, то я видаляю нове поле та виводжу на екран повідомлення про те що таке поле вже існує, це можна помітити на зображанні 3.14.

Лістинг 3.7 – Перевірка на унікальність поля UserID.

```
if (checkIsRepeatValue) {
  if (refPeopleGrid?.current?.dataSource) {
    refPeopleGrid.current.dataSource = (refPeopleGrid.current.dataSource as
          Object[]).splice(1,(refPeopleGrid.current.dataSource as Object[]).length-1);
    People = (refPeopleGrid.current.dataSource as PeopleType[]);
  }
  message.error("This UserID already exists", 2000);
}
```

Також є перевірка на наявність пустого поля. Якщо ми намагаємося додати пусте поле, то у нас нічого не вийде.

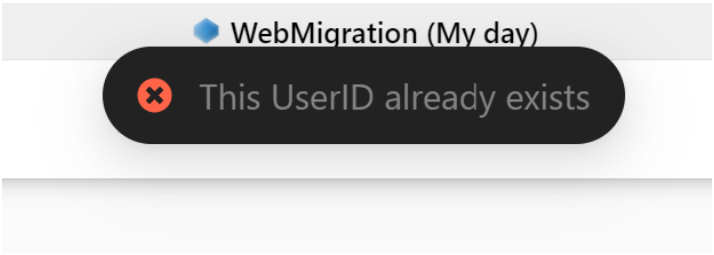


Рисунок 3.14 – Помилка в унікальності нового поля

Завдяки цьому, у нас коректно відображаються данні в таблиці та позначаються де були помилки в додаванні нового поля.

Редагування та додавання відбувається за допомогою модельного вікна, але ми не можемо більше редагувати UserID, ми можемо тільки його вказати в при додаванні нового поля.

Якщо у нас немає помилок в редагуванні та додаванні поля, то відбувається запит до серверу на оновлення даних.

Лістинг 3.8 – Оновлення даних таблиці

```
if (!checkIsIdEmpty && !checkIsRepeatValue) {
  setGridPeople(People);
  setDataAssignedTo (dataAssignedTo);
}
```

Коли користувач намагається видалити поле, тоді з’являється модальне вікно з підтвердженням намірів (рис 3.15).

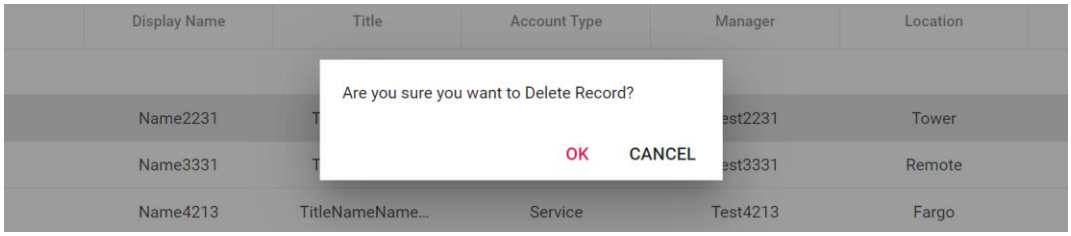


Рисунок 3.15 – Підтвердження намірів щодо видалення рядку

Це зроблено задля запобігання випадкового видалення рядків. Це нормальна практика, бо користувач може випадково натиснути не на ту кнопку.

Хочеться відмітити що поля які знаходяться в модальному вікні для редагування та додавання рядків, не все прості текстові поля. З них: випадаюче меню, лічильник та текстове поле. Ці поля можна помітити на зображенні 3.16.

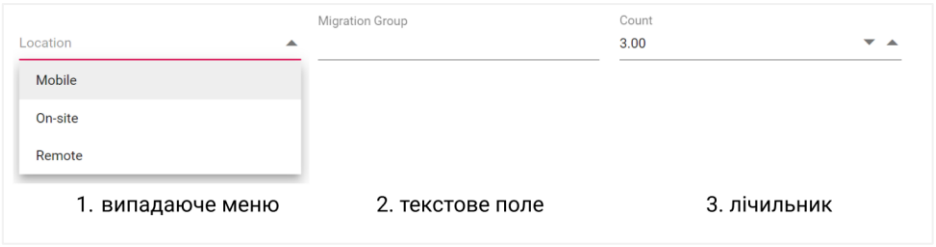


Рисунок 3.16 – Типи форм для заповнення даних

В боковому меню залишилось ти пункти: Mailboxes, Devices та Groups. Всередині цих вкладок, знаходяться таблиці, які пов’язані з основною таблицею people (рис. 3.17).

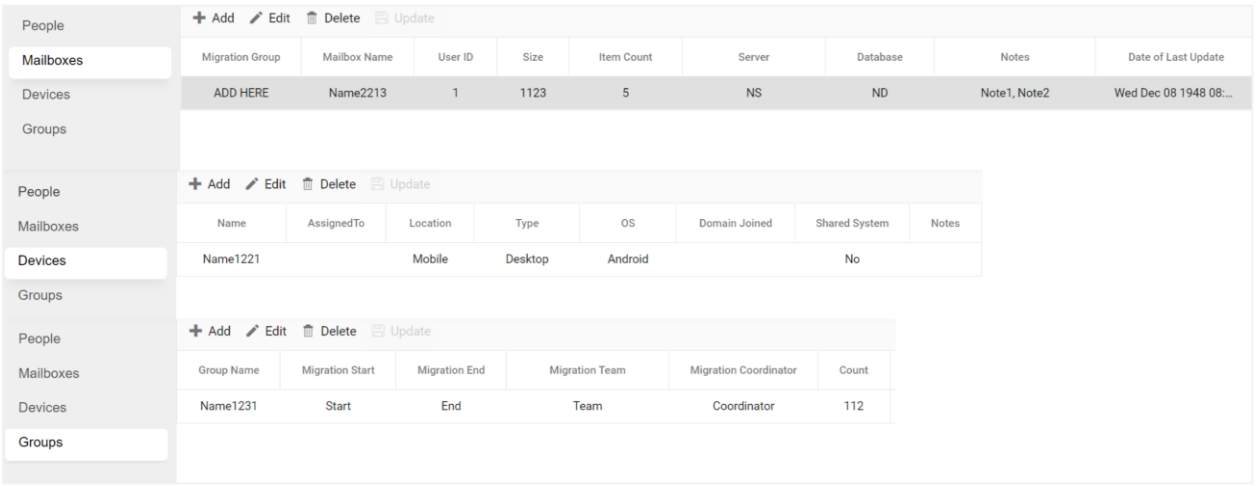


Рисунок 3.17 – Контент вкладок: Mailboxes, Devices та Groups

Коли створюється запис до таблиці Mailboxes або Devices, ми можемо обрати до яких записів People треба додати поточний запис. Це робиться за допомогою випадаючого меню, яке динамічно оновлюється в залежності того

скільки рядків таблиці People. В цьому меню список всіх UserID таблиці People (рис 3.18).

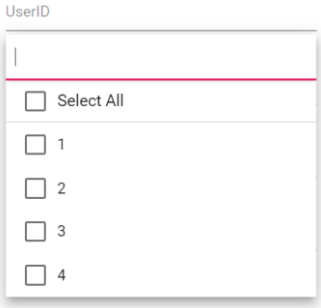


Рисунок 3.18 – Випадаюче меню з вибором записів в People

Якщо поле UserID в Mailboxes або AssignedTo в Devices збігається з полем UserID яке в таблиці People. Тоді поле з Mailboxes або Devices додається в контент поля People.

3.2.4 Вкладка розкладу

Вкладка розкладу слугує для створення розкладу для проекту в простому текстовому форматі як можна побачити на рисунку 3.19.

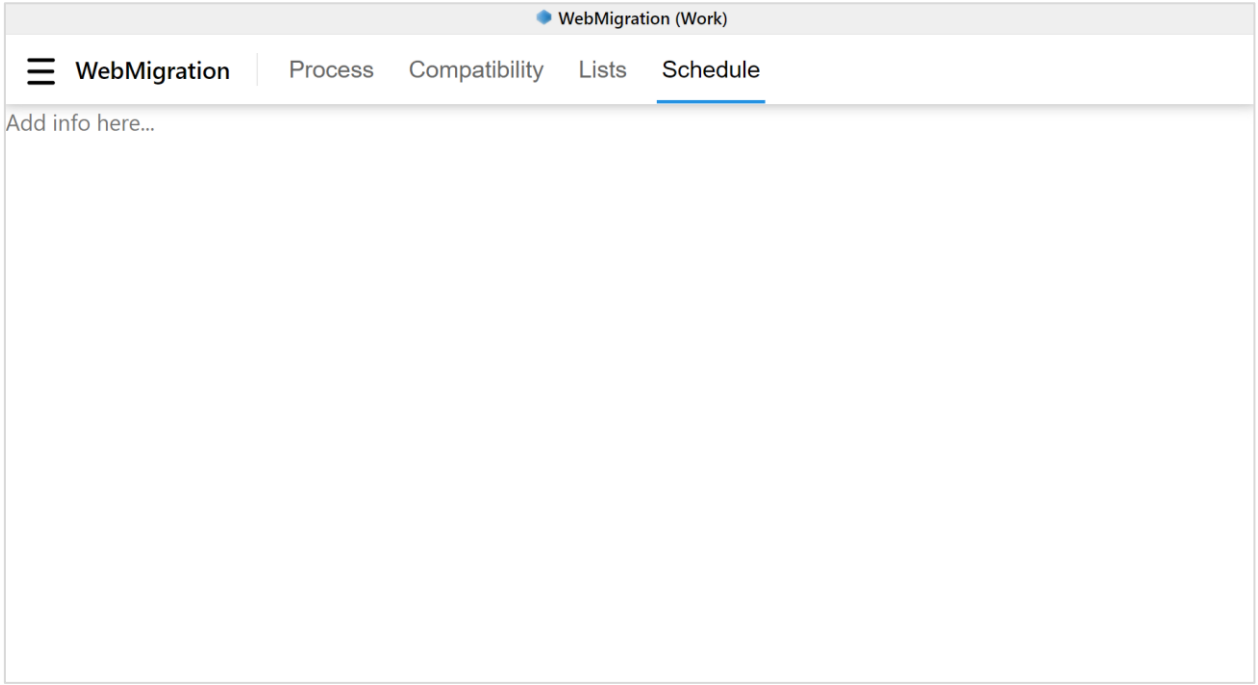


Рисунок 3.19 – Вкладка розкладу.

Для того щоб було можливість щось писати, необхідно додати фрагмент textarea. Textarea – це поле для вводу текстової інформації.

Лістинг 3.9 – Текстове поле для розкладу

```
<Fragment>
    <textarea placeholder="Add info here..." style={{width:"100%", background:"none", border:0, fontSize:17, fontFamily:"Segoe UI", overflow:"auto", height:"calc(100vh - 170px)", resize:"none"}}/>
</Fragment>
```

Цей розклад зберігається на локальному сервері, та слугує орієнтиром для користувача. Хоча цей компонент веб застосунку здається не зручним, але простий текстовий редактор, надає саме лаконічності та простоти. Без зайвих деталей.

3.3 Висновки за розділом

Було розроблено архітектуру веб-застосунку яка складається з різних компонентів. Також розроблено систему створення діаграм та взаємодії між ними. Дизайн був також розроблений за особистим бажанням, він став мінімалістичний, зрозумілий, гарний, простий.

Створений веб-застосунок, повністю виконує свої функції за призначенням. А основною ідеєю було створення зручного і гарного веб-застосунку для покрокового розкладу того як буде відбуватися процес міграції даних, у вигляді діаграм і таблиць.

ВИСНОВКИ

В процесі випускної роботи було виконано всі поставлені задачі щодо створення вебзастосунку з інфорграфічного проектування.

Концепція даного вебзастосунку була рекомендована замовником і є унікальною. Це означає, що існуючих аналогів саме такого продукту не існує, але було розглянуто схожі застосунки за дизайном.

Для розробки продукту було обрано наступний список технологій та засобів: TypeScript, React, Figma, HTML, CSS, Syncfusion, які були регламентовані замовником.

У підсумку було створено зручний та зрозумілий користувачам вебзастосунок, що може бути використаний IT-фахівцями для наглядної демонстрації процесів проектування, у тому числі й міграції великих обсягів даних.

Створений вебзастосунок у повному обсязі задовільнив вимоги технічного завдання замовника. Також можливий подальший розвиток даного проекту та планується перехід з локальних рішень до використання хмарних сервісів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Веб-застосунок. Вікіпедія. [Електронний ресурс]. – Режим доступу: www. URL: https://uk.wikipedia.org/wiki/Вебзастосунок (дата звернення: 29.03.22)

2. Дохід українських фрілансерів на міжнародних ринках зріс на 20% [Електронний ресурс]. – Режим доступу: www. URL: https://biz.nv.ua/ukr/markets/hto-z-ukrajinciv-naybilshe-zaroblyaye-na-mizhnarodnih-rinkah-novini-ukrajini-50202014.html (дата звернення: 19.05.22)

3. Жизненный цикл компонента. [Електронний ресурс]. – Режим доступу: www. URL: https://metanit.com/web/react/2.6.php (Дата звернення: 11.05.22)

4. Основні бібліотеки та фреймворки JavaScript, які вам потрібно знати. [Електронний ресурс]. – Режим доступу: www. URL:https://code.tutsplus.com/ru/articles/essential-javascript-libraries-and-frameworks-you-should-know-about--cms-29540 (дата звернення: 31.03.22)

5. Просторова організація бізнесу в регіонах України: форми та механізми регулювання. [Електронний ресурс]. – Режим доступу: www. URL: http://ird.gov.ua/irdp/p20190004.pdf (дата звернення:21.04.22)

6. Рейтинг мов програмування 2022. C# обійшов Java, TypeScript зрівнявся з PHP, а Dart — найбільш комфортна мова [Електронний ресурс]. – Режим доступу: www. URL: https://dou.ua/lenta/articles/language-rating-2022/ (дата звернення:13.05.22)

7. Рівень безробіття в Україні. Статистика трудових ресурсів України. [Електронний ресурс]. – Режим доступу: www. URL: https://index.minfin.com.ua/labour/unemploy/ (дата звернення: 03.05.22)

8. Скільки заробляють в Україні онлайн (інфографіка). [Електронний ресурс]. – Режим доступу: www. URL: https://news.finance.ua/ru/news/-

/505604/skolko-zarabatyvayut-v-ukraine-onlajn-infografika (дата звернення: 02.03.22)

9. Скільки заробляють IT фрілансери в Україні. [Електронний ресурс]. – Режим доступу: www. URL: https://ubr.ua/labor-market/ukrainian-labor-market/frilansery-nachali-zarabatyvat-bolshe-skolko-i-komu-platjat-3890932 (дата звернення: 10.05.22)

10. Україна посіла 11 місце у Топ-50 країн з найкращими програмістами [Електронний ресурс]. – Режим доступу: www. URL: https://itc.ua/news/reyting-hackerrank-ukraina-zanyala-11-mesto-v-top-50-stran-s-luchshimi-programmistami/ (дата звернення: 15.05.22)

11. Чи є HTML мовою програмування? [Електронний ресурс]. – Режим доступу: www. URL:https://habr.com/ru/post/231897/ (дата звернення:02.03.22)

Що таке TypeScript? [Електронний ресурс]. – Режим доступу: www. URL: http://web.spt42.ru/index.php/chto-takoe-typescript (дата звернення: 01.04.22)

12. Advantages And Disadvantages Of ReactJS [Електронний ресурс]. – Режим доступу: www. URL: https://www.w3schools.blog/advantages-disadvantages-reactjs (дата звернення:18.05.22)

13. Creately - Diagrams & Collaboration – Режим доступу: www. URL: https://chrome.google.com/webstore/detail/creately-diagrams-collabo/mehmcgkakgfcibfkeofncglipefjcfnn?hl=ru (дата звернення: 02.03.22)

14. JavaScript-Глосарій/SEO [Електронний ресурс]. – Режим доступу: www. URL:https://developer.mozilla.org/ru/docs/Glossary/ (дата звернення: 30.03.22)

15. JavaScript-Глосарій/SPA [Електронний ресурс]. – Режим доступу: www. URL:https://developer.mozilla.org/ru/docs/Glossary/ (дата звернення: 29.04.22)

16. React — a JavaScript library for building user interfaces [Електронний ресурс] — режим доступу: https://reactjs.org/ (дата звернення: 20.03.22)

17. React Class Component vs. Functional Component: What's the Difference [Електронний ресурс]. – Режим доступу: www. URL: https://www.telerik.com/blogs/react-class-component-vs-functional-component-how-choose-whats-difference(дата звернення:02.05.22)

18. Single-page application vs. multiple-page application. [Електронний ресурс]. – Режим доступу: www. URL: https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58 (дата звернення: 08.05.22)

19. TypeScript – Overview [Електронний ресурс]. – Режим доступу: www. URL: https://www.tutorialspoint.com/typescript/typescript_overview.htm (дата звернення: 24.05.22)

20. Web API vs REST API Simplified: 4 Critical Differences [Електронний ресурс]. – Режим доступу: www. URL: https://hevodata.com/learn/web-api-vs-rest-api/ (дата звернення:29.05.22)

21. WebStorm Опис. [Електронний ресурс]. – Режим доступу: www. URL: https://itpro.ua/product/jetbrains-webstorm/?tab=description (дата звернення: 06.04.22)

22. Wikipedia: SmartDraw. [Електронний ресурс]. – Режим доступу: www. URL: https://en.wikipedia.org/wiki/SmartDraw (дата звернення: 06.04.22)

23. What is IIS? [Електронний ресурс]. – Режим доступу: www. URL: https://stackify.com/iis-web-server/ (дата звернення:31.05.22)

24. What is .NET? An Overview of the Platform [Електронний ресурс]. – Режим доступу: www. URL: https://auth0.com/blog/what-is-dotnet-platform-overview/ (дата звернення: 29.05.22)

26. What is SQL Server Express? Definition, Benefits, and Limitations of SQL Server Express [Електронний ресурс]. – Режим доступу: www. URL: https://stackify.com/what-is-sql-server-express/ (дата звернення:02.06.22)

ДОДАТОК А

```
import React, {FC, useEffect, useState} from "react";
import {message} from 'react-message-popup'
import {DateTimePickerComponent} from '@syncfusion/ej2-react-calendars';
import {DropDownListComponent} from '@syncfusion/ej2-react-dropdowns';
import {TextBoxComponent} from '@syncfusion/ej2-react-inputs';
import {DialogComponent} from '@syncfusion/ej2-react-popups';
import "./styles/DiagramStyle.css";
import {AccordionComponent, AccordionItemDirective, AccordionItemsDirective} from '@syncfusion/ej2-react-
navigations';
import {
    AnnotationConstraints,
    BpmnDiagrams,
    ConnectorModel,
    DiagramComponent,
    DiagramContextMenu,
    FlowShapeModel,
    Inject,
    NodeConstraints,
    NodeModel,
    PointPortModel,
    PortVisibility,
    ShapeAnnotationModel,
    SnapConstraints,
    UmlActivityShapes
} from "@syncfusion/ej2-react-diagrams";

import {MenuEventArgs} from "@syncfusion/ej2-navigations";
// @ts-ignore
import {
    HtmlEditor,
    Image,
    Inject as RichInject,
    Link,
    QuickToolbar,
    RichTextEditorComponent,
    Toolbar as RichToolbar,
    ToolbarType
} from '@syncfusion/ej2-react-richtexteditor';
// @ts-ignore
import AddIcon from "../../res/img/add.png";
// @ts-ignore
import ProcessesIcon from "../../res/img/Processes.png";
// @ts-ignore
import ArrowIcon from "../../res/img/Arrow Bold.png";
// @ts-ignore
import CloseIcon from "../../res/img/close.png";
// @ts-ignore
import BinIcon from "../../res/img/bin_red.png";
// @ts-ignore
import PreMigrationIcon from "../../res/img/PreMigration.png";
// @ts-ignore
import MigrationIcon from "../../res/img/Migration.png";
// @ts-ignore
import PostMigrationIcon from "../../res/img/PostMigration.png";
```