

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ ДЛЯ ГЕНЕРУВАННЯ
СЕРТИФІКАТІВ ДЛЯ УЧАСНИКІВ МІЖНАРОДНИХ ПРОЄКТІВ
WEB APPLICATION DEVELOPMENT FOR GENERATING THE
CERTIFICATES FOR THE PARTICIPANTS OF INTERNATIONAL PROJECTS

Виконав: студент 4 курсу, групи КНТ-118 сп
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Цибульський О.Г.

(прізвище та ініціали)

Керівник Шитікова О.В.

(прізвище та ініціали)

Рецензент Морщавка С.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 20 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

Цибульського Олега Геннадійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розроблення вебзастосунку для генерування сертифікатів для учасників міжнародних проєктів. Web application development for Generating the Certificates for the Participants of International Projects

керівник проєкту (роботи) Шитікова О.В., к.т.н, доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проєкту (роботи) 21 травня 2021 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Реферат. Перелік скорочень та умовних познач. Вступ. 1. Аналіз предметної області. 2. Вибір та обґрунтування структури системи, яка проєктується. 3. Основні рішення щодо реалізації компонентів системи. 4. Керівництво програміста. 5. Керівництво оператора. Висновки. Перелік джерел посилання. Додатки.

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

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Шитікова О.В., к.т.н., доцент		
Нормоконтроль	Камінська Ж. К., асистент		

7. Дата видачі завдання “25” березня 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2-3 тижні	Розділ 1
3	Розробка архітектури програми	4 тиждень	Розділ 2
4	Розробка програми	5-6 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми	7 тиждень	Розділи 4-5
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування.	8 тиждень	Додатки
7	Захист роботи	9 тиждень	

Студент

(підпис) Цибульський О.Г.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Шитікова О.В.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 95 с., 29 рис., 7 табл., 4 дод., 19 джерел.

СЕРТИФІКАТИ МІЖНАРОДНИХ ПРОЕКТІВ, СИНТАКСИЧНИЙ МЕТОД, ГЕНЕРУВАННЯ, СИНТАКСИС, ДОКУМЕНТ, JAVASCRIPT.

Об'єкт дослідження – процес автоматичної генерації документів.

Предмет дослідження – вебзастосунок для генерування сертифікатів.

Мета роботи – покращення адміністративних процесів керування міжнародними проектами за рахунок генерування відповідних грамот та сертифікатів у форматі PDF документів.

Актуальність полягає в тому, що більшість міжнародних проектів та організацій зустрічаються з проблемою створення та заповнення великої кількості екземплярів сертифікатів для всіх учасників програм та подій. Чим більша кількість учасників, тим більше часу виділяється для створення та заповнення сертифікатів власноруч. Це призвело до необхідності повної автоматизації заповнення сертифікатів за короткий термін часу, не зважаючи на кількість учасників.

В роботі наведено інформацію щодо методу шаблонізації, оброки та генерування PDF документів. Даний метод полягає у розміщенні, аналізі та обробці компонентів шаблону та подальшого генерування сертифікатів у відповідності до даних учасників.

Розроблено програмний продукт, який призначений для вирішення завдання автоматизації генерування сертифікатів, а саме створення текстового шаблону та його використання для динамічного генерування сертифікатів у відповідності до даних учасників.

ABSTRACT

Explanatory note to the final qualifying work of the bachelor: 95 pages, 29 figures, 7 tables, 4 appendixes, 19 sources.

CERTIFICATES OF INTERNATIONAL PROJECTS, SYNTAX METHOD, GENERATION, SYNTAX, DOCUMENT, JAVASCRIPT.

The object of research is the process of automated generation of documents.

The subject of the research is a web application for generating certificates.

The purpose of the work is to improve the administrative processes of international project management by generating the relevant diplomas and certificates in PDF format.

The relevance is that most international projects and organizations face the problem of creating and filling a large number of copies of certificates for all participants in programs and events. The larger the number of participants, the more time is allocated to create and complete all certificates. This has led to the need to fully automate the completion of certificates in a short period of time, regardless of the number of participants.

The paper provides information on the method of templating, process and generating of PDF documents. This method is about placing, analyzing and processing the components of the template and further generating certificates in accordance with the data of the participants.

A software product has been developed to solve the problem of certificate automated generation process, creation of a text template and its usage for dynamic certificate generation in accordance with the data of participants.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Основні поняття	11
1.2 Характеристика процесу генерування електронних документів	13
1.3 Порівняльний аналіз існуючих рішень генерування електронних документів.....	14
1.4 Постановка завдання до роботи.....	15
1.5 Висновок	17
2 Вибір та обґрунтування структури системи, яка проєктується.....	18
2.1 Вибір інструментарію розробки програмного продукту	18
2.2 Структурна схема програмного продукту	26
2.3 Висновок	30
3 Основні рішення щодо реалізації компонентів системи.....	31
3.1 Алгоритм вводу даних файлів до системи	32
3.2 Використання синтаксичного методу модулем обробки даних.....	33
3.3 Процес повної генерації та збереження документу.....	35
3.4 Висновок	36
4 Керівництво програміста.....	37
4.1 Призначення й умови використання	37
4.2 Характеристики програмного продукту	37
4.3 Звернення до програмного продукту	38
4.4 Вхідні і вихідні дані	39
4.5 Повідомлення	42
4.6 Висновок	42
5 Керівництво оператора	43
5.1 Призначення й умови виконання програмного продукту.....	43
5.2 Виконання програмного продукту	43

5.3 Повідомлення оператору	50
5.4 Висновок	50
Висновки	51
Перелік джерел посилення	53
Додаток А Технічне завдання	55
Додаток Б Опис програмного продукту	61
Додаток В Текст програми	72
Додаток Д Слайди презентації.....	89

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interface;
IDE	– Integrated Development Environment;
PDF	– Portable Document Format;
WEB	– веб;
B3	– вебзастосунок;
ІС	– інформаційна система;
КТШ	– компонент текстового шаблону;
ПП	– програмний продукт;
ТШ	– текстовий шаблон;

ВСТУП

На сьогоднішній день існує велика кількість міжнародних проектів та організацій, що заохочує велику кількість студентів та спеціалістів до участі в їх проектах, змаганнях та офіційних подіях. У більшості таких подій, кожному учаснику видається грамота або сертифікат, що засвідчують факт участі або посідання призових місць та отримання відзнак. Це веде за собою доволі тривалу роботу по організації процесу створення сертифікаційних документів для кожного з учасників.

Під створенням сертифікаційних документів мається на увазі повна розробка документу. Цей процес включає в себе створення дизайну (підбір розміру документу, основного фону, базових кольорів, створення рамки, розміщення емблем), оформлення тексту (підбір розміру тексту, стилю, кольору), розміщення областей та їх заповнення для тексту з інформацією кожного з учасників та подальший друк кожного з екземплярів.

Заповнення сертифікатів, за першого погляду, здається доволі простим процесом, що не повинен виконуватися протягом тривалого часу. Насправді час, що витрачається на заповнення всіх документів, прямо пропорційно залежить від кількості учасників, що приймали участь у проектах, подіях або змаганнях. Інакше кажучи, чим більше людей приймає участь у певному проекті, тим більше часу витрачається на створення загальної кількості документів.

Для вирішення цієї проблеми було розроблено ВЗ призначений для автоматизації процесу заповнення сертифікаційних документів. Було виконано дослідження та виконано програмну реалізацію методу генерації сертифікатів

Розроблено ВЗ для рішення задачі автоматизації створення PDF документі, що містять у собі дані у вигляді оформленого сертифікату, кожен з яких заповнений даними учасників на основі розробленого методу. Розроблений метод вирішує питання динамічної підстановки даних учасника

до сертифікаційного документу за допомогою використання синтаксису, що аналізується розробленою системою під час збереження файлу.

Отримані результати надають зручний метод для оптимізації роботи по заповненню сертифікаційних документів та винагород, що може бути використаний для різні сертифікати та документи будь-якої організації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні поняття

Проект – це тимчасове підприємство, яке спрямоване на організацію особистої події, метою якої є надання послуг, можливостей, створення унікального продукту та створення умов обміну знаннями та ідеями. Кожен з проектів має свою характеристику, за допомогою якої можна чітко визначити чи є організована подія проектом. Серед основних характеристик виділяють:

- тимчасовість;
- унікальність продукту проекту;
- послідовність організації проекту [1].

Характеристика тимчасовості проекту вказує на те чи має організована подія чіткі дати початку та кінця події. У разі якщо дана подія не має визначеного терміну проведення, то її неможна називати проектом [1].

Характеристика унікальності продукту проекту вказує на унікальність результату, розробки або дослідження проекту. Інакше таку подію можна назвати «серійним виробництвом» [1].

Характеристика послідовності організації проекту вказує на строгий розвиток та проведення проекту згідно з етапами, що були чітко сформульовані та описані у створених на початковому етапі специфікаціях проекту [1].

Не зважаючи на унікальність продукту кожного з проектів, усі вони мають спільний підхід до реалізації, а саме:

- організовується та виконується чітким складом людей, кожен з яких є представником певних громад та юридичних осіб, що являються ініціаторами, співорганізаторами та спонсорами проекту;

- має чіткі обмеження по ресурсам та фінансуванню, що виділені для реалізації продукту проекту;

– планується, організовується та керується представниками організації-ініціатора, враховуючи ідеї та думки співорганізаторів та спонсорів проекту [1].

Міжнародний проект – це проект, в організації та керуванні якого беруть участі представники з двох чи більш країн і які залучають велику кість студентів та спеціалістів до участі з усього світу.

Сертифікат – документ, що свідчить про те, що людина або підприємство, мають певний рівень кваліфікації. Це надає можливість володарю сертифіката виконувати професіональну діяльність у певній сфері [2].

Грамота – офіційний акт правового значення, документ, що свідчить або встановлює що-небудь. Є документом, яким нагороджують за успіхи та відзнаки у будь-яких видах офіційної діяльності [3]–[4].

Генерування документу – процес покрокового створення екземплярів електронного документа, кожен з яких буде містити як загальну так і унікальну для кожного з екземплярів інформацію. Результатом даного процесу повинен бути документ, що містить у собі усі створені екземпляри, або папку, в якій кожен окремий файл є окремим екземпляром документа-оригіналу, що являє собою електронний документ.

Електронний документ - документ, інформація в якому зафіксована у вигляді цифрових даних, містить детальну інформацію про тип документу. Електронний документ може бути створений, переданий, збережений і перетворений електронними засобами у візуальну форму, за допомогою якої виконуватиметься відображення даних, які він містить. Ці дані можуть бути відображеними як на моніторі, так і роздруковані на папері [5].

Структурований документ – це електронний документ, в якому використовується метод розмітки для ідентифікації усього або декількох частин документа, що використовується для визначення статичної та динамічної частин документа і подальшої його заповнення необхідними даними.

1.2 Огляд існуючих методів створення документів

На сьогоднішній день робота з системами автоматизації роботи по редагуванню та створенню електронних документів є доволі складно, не зважаючи на вже реалізовані системи, що надають можливості по роботі з файлами.

Серед більшості з них можна виділити два основних методи створення документів:

- синтаксичний метод;
- за допомогою структурного документу.

Основний принцип синтаксичного методу полягає у завантаженні документа-шаблону до ІС, його обробкою та визначенню частин документа, що повинні бути замінені динамічними даними. Після аналізу виконується етап обробки, де згідно с проаналізованими даними виконується заміщення шаблонізованих фрагментів на відповідні дані, після чого буде отримано повністю готовий документ.

Основний метод використання структурного документу полягає у завантаженні структурного документа типу XML, що містить у собі повні інформацію про розмітку та склад документу. Далі необхідно виконати процес компіляції структурного файлу у дані, зрозумілі текстовому редактору, визначення статичних та динамічних фрагментів документу, та зробити віртуальну копію документу для обробки. Процес обробки віртуальної копії документу має такий же алгоритм, що й принцип синтаксичного методу. Після повної обробки віртуальної копії документу виконується цифровий друк отриманих даних до електронного документу та його подальше збереження до файлової системи комп'ютера.

Особливості кожного з методів наведено у таблиці 1.2.

Таблиця 1.1 – Порівняння особливостей синтаксичного методу та методу використання структурного файлу

Особливість	Синтаксичний	Використання структурного файлу
Простота використання	+	–
Необхідність додаткової компіляції	–	+
Швидкість обробки	+	±
Гнучкість використання	+	+
Надійність стійкості шаблону	–	+

Згідно з табл. 1.1 усі перелічені методи мають спільний алгоритм створення екземплярів документу з методом підстановки динамічних фрагментів відповідними з ІС. Але є певні відмінності.

Для алгоритму з використанням структурного файлу необхідно пройти етап компіляції XML файлу, що робить необхідним використання сторонніх модулів та бібліотек для виконання цієї операції. Також даний метод є складним для зрозуміння звичайному користувачу, що не знайомий з файлами гіпертекстової розмітки та принципами роботи з ними.

Для синтаксичного алгоритму не потрібні додаткові обробки та компіляції, так як ми вже маємо шаблон в ІС. Але використання цього методу у багатьох рішеннях є або дорогим, або ускладненим великої кількістю додаткових алгоритмів, що використовуються для переорганізації документу, контроль алгоритму виконання та додаткового адміністрування процесу.

1.3 Огляд існуючих сервісів створення та обробки документів

На сьогоднішній день вже розроблено вебсервіси для надання можливості створення та редагування документів. У табл. 1.2 наведено одні з доступних вебсервісів, котрими користується достатньо широке коло користувачів.

Таблиця 1.2 – Порівняльна характеристика вебсервісів по редагуванню документів

Характеристика	PDF24 [6]	PDF2GO [7]	Microsoft Word Online [8]
Широкий вибір формату обробки	±	±	++
Локалізація інтерфейсу на різні мови	+	+	+
Безкоштовний доступ до інструментів редагування	++	±	++
Можливість редагувати різні формати документів	±	±	++
Можливість конвертації один формат документів в інший	++	+	—
Інструменти генерування документів по шаблону	—	—	—
Комп'ютерна версія ВЗ	+	—	+

Згідно з табл. 1.2 можна зробити висновок, що обрані вебсервіси задовольняють виконання більшості необхідних для редагування документів завдань.

Кожна з вище зазначених платформ веде свою політику доступу до власних інструментів, але жоден з них не надає можливість генерувати документи з використанням шаблону та даних, завантажених до ІС. Це надає підставу для створення інструмента, що надає можливість використовувати процес генерування документів.

1.4 Постановка завдання до роботи

В даній роботі досліджується проблема автоматичного створення сертифікатів міжнародних проектів.

В розроблюваному ПЗ для автоматизації процесу створення документів необхідно створити загальний ТШ, що складається з КТШ. Кожен з цих компонентів задаються на площині документу, що являє собою площину з декартовою системою координат $\{x_i, y_i\}$, $i = \overline{1, m}$. Сукупність створених КТШ становить множину $I = \{1, 2, \dots, m\}$. Використовуючи таку систему можна бути впевненим у чіткому розміщені тексту на площині документу, що оброблюється. В свою чергу, кожен КТШ містить у собі текст, що у собі статичні та динамічні дані щодо текстових фрагментів, сукупність яких являє собою лінійну структуру $\{x_j\}$, $j = \overline{1, n}$. Загальну множину текстових фрагментів КТШ можна описати множиною $J = \{1, 2, \dots, n\}$.

За отриманими даними можна зробити висновок, що розроблюваний алгоритм під час обробки ТШ виконує певну кількість операцій для виконання обробки та друку одного екземпляру документу. Загальна кількість операцій $P_{sum.op}$ для обробки одного екземпляру розраховується за формулою (1.1):

$$P_{sum.op} = \sum_{i=1}^m \sum_{j=1}^n P_{ij} \quad (1.1)$$

де $P_{sum.op}$ – загальна кількість операцій;

P_{ij} – операція обробки фрагменту тексту КТШ.

Метою роботи було створення ПП для автоматизації процесу генерування сертифікатів шляхом використання синтаксичного методу обробки ТШ та фрагментів тексту його компонентів.

Для виконання поставленої задачі необхідно виконати наступні завдання:

- створити метод для завантаження та відображення документу в ІС;
- створити метод для завантаження та перегляду даних текстового CSV файлу;

- створити метод для створення та організації ТШ для завантаженого документу;
- створити метод відображення ТКШ в області відображення завантаженого документу;
- створити метод обробки тексту КТШ, з використанням синтаксичного методу обробки;
- створити метод відображення кожного із згенерованих екземплярів документа;
- створити метод завантаження усіх згенерованих екземплярів документа у пам'яті комп'ютера.

1.5 Висновок

В цьому розділі було досліджено предметну область, визначено основні поняття і терміни, пов'язані з предметною областю, дано характеристики ПП для автоматизації генерування сертифікатів.

Було проаналізовано основні методи вирішення завдання та наведено порівняльний аналіз цих методів, за результатом якого було виявлено перевагу синтаксичного методу як для розробки, так і для розширення можливостей інструменту.

Проведено порівняльний аналіз ВЗ, призначених для створення та редагування документів, виявлено переваги та недоліки кожного із них.

Було сформульовано і поставлено завдання до робіт.

2 ВИБІР ТА ОБҐРУНТУВАННЯ СТРУКТУРИ СИСТЕМИ, ЯКА ПРОЄКТУЄТЬСЯ

2.1 Вибір інструментарію розробки програмного продукту

Одним із основних етапів життєвого циклу ІС та ПП є вибір інструментів розробки та технологій, що повинні бути підключені та реалізовані в ПП.

Основна задача ПП – це програмна реалізація методу генерування грамот та сертифікатів у форматі PDF документів.

Розроблюваний ПП повинен обов'язково містити такі модулі:

- центральний модуль;
- зберігання даних;
- вводу-виводу даних;
- редагування;
- обробки.

Перед початком проектування ПП необхідно обрати такі інструменти, як мова програмування, середовище розробки (Integrated Development Environment – IDE) та набір необхідних бібліотек та API для обробки даних, що введені до ПП.

2.1.1 Вибір мови програмування

Для вирішення поставленої задачі, яка полягає в розробці ВЗ для реалізації процесу генерування сертифікатів необхідно зробити огляд сучасних мов програмування, що використовуються для розробки ВЗ та порівняти їх одна з одною, щоб обрати її для реалізації поставленої задачі. Нижче наведемо основну інформацію щодо найбільш популярних мов програмування для розробки ВЗ, їх переваги та недоліки.

JS (JavaScript) – скриптова мова програмування, що використовується для створення одно сторінкових ВЗ, написання програм для роботи сервера,

мобільних застосунків, створення сценаріїв взаємодії користувача з інтерфейсом ВЗ на боці клієнта, а саме: керувати подіями браузера, змінювати структуру та відображення інтерфейсу, обмінюватись даними з сервером в асинхронному режимі [9].

JS класифікують як скриптову мову програмування з динамічною типізацією, що підтримує прототипну, імперативну та функціональну парадигми програмування. Цій мові також властиві відповідні архітектурні особливості, а саме: динамічна типізація, використання функцій як об'єктів першого класу, прототипне наслідування та автоматичне керування пам'яттю [9].

Основні переваги JS:

- дана мова підтримується усіма існуючими браузерами;
- мова є дуже універсальною, так як використовується для створення клієнтських та серверних ПП;
- є дуже популярною мовою програмування ВЗ. Про це свідчить велика кількість ресурсів, що пропонують інформацію про вивчення мови. Також на таких платформах, як StackOverflow та GitHub, можна знайти велику кількість обговорень для проектів, що пов'язані з мовою програмування JS;
- велика кількість готових рішень, бібліотек та фреймворків, які можна використовувати для реалізації широкого спектру задач.

Основні недоліки JS:

- можливість виявити помилки тільки після компіляції та виконання ПП;
- можливості самої мови дуже обмежені, що призводить до необхідності використання сторонніх бібліотек для фреймворків.

Python – об'єктно-орієнтована мова програмування з динамічною типізацією. Дана мова використовується як основна при роботі з головним WEB фреймворком, що пов'язаний за цією мовою, Django. В основному мова Python використовується для написання серверних програм та програм адміністрування вебсторінок [10].

Python підтримує модульну роботу та процеси взаємодії, що надає можливість структурувати проекти по модулям і надає можливість повторно використовувати розроблені модулі та код. Ця мова підтримує наступні парадигми програмування: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована [10].

Основні переваги Python:

- синтаксис мови легкий та швидкий для розуміння контексту виконання коду;
- інтерпретатор Python реалізований на більшості відомих платформах та операційних системах;
- наявність великої кількості модулів, що вирішують більшість необхідних задач та завдань.

Основні недоліки Python:

- невисока швидкість виконання коду програми, що пов'язано з методом інтерпретації коду;
- відсутня строга типізація даних, що призводить до використання більшої кількості пам'яті;
- блокування потоку інтерпретатора GIL (Global Interpreter Lock), що пов'язане з однопотомним процесом обробки складних типів даних (словники, колекції) і має вплив на швидкість виконання програми.

PHP – скриптова мова програмування, що дозволяє доволі швидко і легко масштабувати ВЗ. Призначена для написання динамічних HTML-сторінок на боці WEB сервера, в наслідок чого вони відображається на стороні клієнта. Особливістю даної мови є можливість використовувати гіпертекстову мову HTML для написання сторінки і вбудовувати PHP код всередині тегу «<?php >» всередині HTML розмітки [11].

Основні переваги PHP:

- вбудовані бібліотеки та інтерфейси для взаємодії з різними типами баз даних: MySQL, PostgreSQL, SQLite, mSQL, Oracle та ін.;
- синтаксис мови дуже схожий до мови C та Perl, що значно спрощує розробку для розробників, що мають досвід роботи з цими мовами;
- є достатньо популярною мовою серед широкого кола розробників, про що свідчить велика кількість бібліотек та фреймворків реалізованих за допомогою PHP;
- виконання коду виконується доволі швидко, у порівнянні з іншими мовами призначеними для WEB програмування.

Основні недоліки PHP:

- складний процес створення імен функцій, що призводить до надто довгих імен функцій і плутанини при перевірці коду;
- швидкість PHP здобувається завдяки процесу кешування даних та операцій, що призводить до більшого використання пам'яті [11].

Кожна з вищеписаних мов програмування має свої переваги та можливості, що можуть бути використані для програмної реалізації генерування сертифікатів. Завдяки даним мовам програмування здійснюється збереження, аналіз та обробка даних та подій, що викликані користувачем у ВЗ. Слід зазначити, що кожна з цих мов активно розвивається та реалізує нові можливості для роботи та аналізу даних, спрощує та вдосконалює вже існуючі методи та доповнюється новими бібліотеками та рішеннями.

Під час збору інформації про мови програмування було створено порівняльну характеристику мов програмування, що наведена у табл. 2.1.

Таблиця 2.1 – Порівняльна характеристика мов програмування

Критерії порівняння	JavaScript [9]	Python [10]	PHP [11]
Мова створювалася спеціально для веброзробки	+	—	+
Кросплатформеність	+	+	+
Швидка інтерпретація коду	++	—	+
Наявність зрозумілої та повної документації	+	+	+
Простота вивчення та розробки	—	++	+
Виробничість(продуктивність роботи)	+	—	++
Наявність бібліотек та фреймворків з повною документацією	+	+	—
Наявність великої кількості методів обробки даних	+	++	+
Паралельне програмування	+	+	+
Можливість інтеграції та взаємодії з іншими мовами	++	+	+

Як видно з табл. 2.1 кожна з мов програмування має свої переваги та недоліки, але для реалізації програмного рішення щодо генерування сертифікатів, була обрана мова JavaScript. Мова JavaScript має велику кількість інструментів та бібліотек з відкритим вихідним кодом, що дозволяє легко досліджувати процеси обробки та знаходити необхідні параметри та методи для вирішення певних задач. Ця мова широко поширена серед розробників і підтримується усіма WEB браузерами без винятку, що робить усі ВЗ, що були розроблені з використанням мови JavaScript, доступними до більшості клієнтів та користувачів, їх доволі легко підтримувати та поширювати, що є одним із головних факторів розрахунку вартості розробки проекту. Враховуючи все вище вказане, для розробки ПП було обрано мову програмування JS.

2.1.2 Вибір середовища розробки

Для розробки даного ВЗ необхідно обрати середовище розробки, що надає можливості швидко писати код ПП.

Для роботи з мовою JavaScript існують такі середовища розробки: IntelliJ IDEA, Visual Studio Code, Sublime Text, Notepad++, Vim.

Кожне з вище названих середовищ надає можливість писати ПП використовуючи мову JavaScript. Але найбільшу популярність здобули WebStorm та Visual Studio Code – IDE з широким інструментарієм та процесами покращення та полегшення написання, компіляції та моніторингу виконання коду.

WebStorm – середовище розробки, що надає можливість працювати з мовами JavaScript, HTML, CSS та підтримує швидко розробку та інструменти автоматизації створення базової структури проектів для популярних фреймворків мови JavaScript, а саме: Angular, ReactJS, Vue.js, Node.js та ін. Вміщує у собі усі необхідні інструменти для швидкого та комфортного написання коду на мовах HTML, CSS, JavaScript [12].

WebStorm доступний тільки у платній версії, що надає підтримку роботи не тільки з мовою програмування JavaScript та його фреймворками, але також має інструменти для роботи та взаємодії з базами даних, системою контролю версій, модулі тестування. Також надає можливість компілювати та виконувати написаний код, надає доступ до консолі системи для взаємодії з пакетами та інструкціями програми [12].

Середовище розробки WebStorm доступна для роботи на наступних платформах: Windows, MacOS, Linux.

Основні переваги WebStorm:

- швидке автоматичне створення коду у відповідності до контексту;
- вбудований компілятор;
- інтеграція з інструментами створення та обробки структури проектів, такі як Apache та Webpack;

- вбудована система контролю версій ПП із можливістю синхронізації з сервісом GitHub [12].

Основні недоліки WebStorm:

- велика кількість інструментів, для використання яких необхідно вивчити принципи їх роботи, ознайомитись з об'ємною документацією, що достатньо важко для розробників початківців та нових користувачів IDE взагалом;

- можливість використовувати IDE у тестовому режимі лише 30 днів з моменту встановлення ПЗ у систему. Після закінчення терміну безкоштовного використання необхідно обов'язково придбати ліцензію для подальшої роботи з IDE [12].

Visual Studio Code – доступне та легке IDE з відкритим вихідним кодом, що надає можливість користувачам розробляти та доповнювати функціонал даного середовища розробки, що підтримує роботу з більшістю відомих мов, серед яких: C, C++, C#, Python, PHP, JavaScript, TypeScript [13].

Саме по собі це IDE не має широкого інструментарію, у порівнянні з WebStorm, але є легшим інструментом для сприйняття та подальшої роботи. Крім того є можливість встановлення додаткових модулів та плагінів, що можуть надати нові можливості та функціонал, спростити процес написання та редагування коду. Редактор також має підтримку технології IntelliSense, що надає можливість швидко писати та доповнювати код у відповідності до контексту коду [13].

Visual Studio Code знаходиться у вільному доступі без обмежень, що робить її однією з найпоширеніших та найпопулярніших у використанні серед всіх WEB розробників.

Visual Studio Code доступний для роботи на наступних платформах: Windows, MacOS, Linux.

Основні переваги Visual Studio Code:

- зрозумілість та легкість інтерфейсу та відсутність нагромадження великої кількості інструментів;

- можливість самостійно налагодити середу розробки та встановити необхідні модулі для редагування, перевірки коду та компіляції коду;
- інтегрована система компіляції коду, контролю версій та взаємодія з сервісом GitHub;
- безкоштовний доступ до основних можливостей IDE з можливістю самостійного налагодження та розширення можливостей редактору [13].

Основні недоліки Visual Studio Code:

- дуже малий набір інструментів на початку роботи;
- необхідність вміти підбирати та налагоджувати необхідні інструменти для вирішення проблем та певних задач [13].

В результаті дослідження вищеописаних було складено порівняльну характеристику середовищ розробки, яка наведена у табл. 2.2.

Таблиця 2.2 – Порівняльна характеристика середовищ розробки

Критерії порівняння	Visual Studio Code [13]	WebStorm [12]
Кросплатформеність	+	+
Доступність	++	±
Наявність автодоповнення	+	+
Наявність автоматичного закриття дужок, брекетів	+	+
Підтримка репозиторіїв	+	+
Наявність «plugin manager»	+	–
Простота встановлення	+	+
Підтримка аналізу коду	+	+
Підтримка рефакторингу коду	+	+
Наявність інтерактивної консолі	+	+
Зручний користувацький інтерфейс	++	±
Швидкість модернізації	++	+

Після перегляду табл. 2.2 можна зробити висновок що найбільші переваги має середовище розробки Visual Studio Code, бо воно має можливість налагоджувати середовище, має вбудований помічник для легкого редагування коду та навігації між залежними файлами, постійно

модернізується та підтримується не тільки розробниками середовища, але й користувачами, має більший рівень доступності. Отже для розробки було обрано середовище Visual Studio Code.

2.2 Структурна схема програмного продукту

Розроблений ВЗ повинен складатися з наступних модулів:

- центральний модуль;
- зберігання даних;
- вводу-виводу даних;
- редагування;
- обробки.

Розроблений ВЗ використовує дані щодо учасників проектів міжнародних організацій та компаній та PDF-документа, що є трафаретом, необхідним для обробки та подальшого генерування сертифікатів на основі даних учасників в якості вхідних даних. Ці дані можуть бути представлені у наступних форматах:

- текстові файли (CSV);
- універсальні документи (PDF).

Дані формати були обрані для стійкості та надійності даних, що завантажуються, зберігаються та оброблюються для генерування сертифікатів.

В якості центрального модуля було розроблено клієнтський ВЗ з використанням бібліотеки ReactJS. Даний модуль відповідає за відображення збудованого графічного інтерфейсу ПП, виконання сценаріїв та подій програми у результаті взаємодії клієнта з активними елементами інтерфейсу.

Бібліотека ReactJS надає можливість створення власної архітектури клієнтської частини проекту, користувацьких компонентів зі створенням подій для оновлення контенту та відображення, що вміщує компонент. Також

є головним вузлом зв'язку сторонніх інструментів, що доповнюють технологію необхідними можливостями та функціями [14].

Для реалізації модуля зберігання даних була застосована технологія React-Redux. React-Redux являється додатковою службою бібліотеки ReactJS, що надає ПП зв'язок з модулем зберігання даних через взаємодію клієнта з інтерфейсом та проходженням компонентів проекту через життєві цикли існування компоненту. Технологія React-Redux може зберігати усі типи даних, що реалізовані у мові JavaScript, від простих (string, number, symbol, boolean) до складних (object, array). Для взаємодії з кожним елементом модуля зберігання даних необхідно створити методи збереження та видачі конкретних фрагментів пам'яті даного модуля. Це значно спрощує процес створення архітектури проекту та надає можливість чітко контролювати процеси обміну даних [15].

Переваги використання технології React-Redux:

- можливість створення власної архітектури організації пам'яті та написання сценаріїв у відповідності до заздалегідь прописаних подій;
- надання доступу до збережених даних у будь якому компоненті проекту;
- легкість у використанні створених сценаріїв та подій [15].

Саме цьому в якості модуля зберігання даних було обрано технологію React-Redux.

В якості модуля вводу-виводу даних застосовуються вбудовані до WEB браузера інструменти взаємодії з файловою системою, що необхідні для вводу CSV списку та PDF документа до IC, та збереження обробленого документу у пам'яті пристрою.

Для виводу даних із CSV документів у інтерфейсі ПП використовуються користувацькі компоненти виводу даних, створених та компільованих за допомогою бібліотеки ReactJS.

Для виводу даних щодо завантаженого PDF документа в інтерфейсі ПП було обрано інструмент React-PDF, що спеціалізується на виведенні

даних щодо завантажених PDF файлів, яке реалізовано на основі взаємодії інструменту з тегом гіпертекстової розмітки HTML «<canvas>». Це надає можливість малювати та редагувати текстові компоненти та шаблонізувати документ для його подальшої обробки [16].

Для створення PDF документу перед завантаженням файлу у пам'ять пристрою було обрано бібліотеку jsPDF, що спеціалізується на створенні, редагуванні та збереженні PDF документів [17].

Для реалізації модуля редагування ТШ було написано форму, реалізовану за допомогою інструменту Redux-Form, та реалізовано алгоритм розміщення КТШ у відповідності до даних з форми. Інструмент Redux-Form надає можливість використовувати окремо виділену область пам'яті, що виділяється за допомогою модуля зберігання даних, для зберігання, редагування та виводу даних без написання методів збереження та отримання введених до форми даних [18].

При кожній зміні значень будь-якої області пам'яті форми відправляється ПП сигнал про зміну, що дає можливість одразу отримати та вивести нові дані без необхідності сповіщення ІС про зміну. Такий підхід надає можливість спостерігати зміни кожного з КТШ, параметри якого були змінені та одразу ж виведені у інтерфейсі ПП.

Для відображення КТШ в інтерфейсі ІС було написано алгоритм, що має доступ до тегу «<canvas>», котрий містить у собі інформацію про завантажений PDF документ. Алгоритм, перебираючи дані про кожний КТШ, друкував введений текст із параметрами, що були введені у формі редагування КТШ.

Для реалізації модуля обробки було використано алгоритм, що працює по принципу алгоритму, що використовується у модулі редагування ТШ. Але окрім розміщення КТШ на документі, цей алгоритм також проводить аналіз та обробку тексту, що був введений у формі.

Текст, що вводиться до ТКШ, поділяється на два типи:

- статичний;

– динамічний.

Статичний текст є незмінним для кожного екземпляру сертифікату та ігнорується алгоритмом обробки тексту.

Динамічний текст – частина тексту КТШ, що має патерн « $\{\text{текст}\}$ ». За допомогою тексту, що знаходиться у фігурних скобка, виконується пошук значення параметру із CSV документу, завантаженого до ІС. Далі замість даного патерну буде виведено значення з CSV документу.

Після процесу обробки, результат обробки кожного екземпляру передається до модуля виводу даних. За допомогою інструменту jsPDF усі передані екземпляри передаються до створеного PDF документу, кожна сторінка якого вміщує окремий екземпляр сертифікату.

На рис. 2.1 зображена структурна схема розробленого ПП.

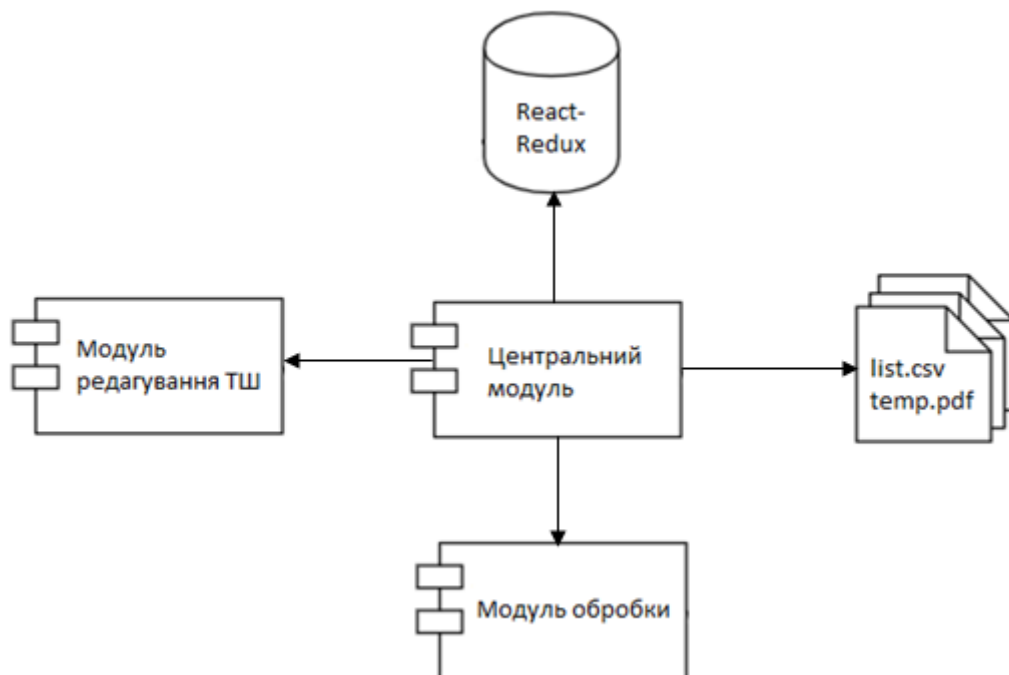


Рисунок 2.1 – Структурна схема програмного продукту

Згідно з рис. 2.1 розроблений ПП використовує дані учасників, що перелічені в CSV файлі, та макет сертифікату у форматі PDF, які опрацьовуються за допомогою спеціальних алгоритмів аналізу КТШ та їх тексту.

Інструмент React-Redux було обрано для тимчасового зберігання даних про учасників, та макет сертифікату. За функцію завантаження необхідних даних до ВЗ відповідає модуль вводу-виводу даних, що базується на вбудованих інструментах WEB браузера для його взаємодії з файловою системою. Модуль редагування використовує дані про КТШ, що зберігаються у спеціальному сховищі модуля зберігання даних, та виконує друк тексту поверх макету сертифікату. Для модуля обробки використовується алгоритм аналізу тексту КТШ, результат обробки якого видається модулю вводу-виводу даних для збереження згенерованих екземплярів сертифікатів у пам'яті комп'ютера. Центральний модуль відповідає за взаємозв'язок усіх вище перерахованих модулів. Даний модуль написаний з використанням бібліотеки React JS, за допомогою якого було розроблено графічний інтерфейс та сценарії взаємозв'язку модулів та користувача з ВЗ.

2.3 Висновок

В результаті виконання роботи по даному розділу було визначено чітку структуру ПП та описано всі модулі, що повинно бути розроблено для реалізації автоматизованої системи генерування сертифікатів. Підібрано необхідний інструментарій, описано принципи їх роботи та процедури, що вони виконують для реалізації поставленої задачі.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

В ході виконання роботи розглядається проблема автоматизації процесу генерування унікальних екземплярів документів. Цю проблему можна вирішити, розробивши алгоритм обробки з використанням синтаксичного методу обробки тексту.

Завдання роботи полягає у розробці вебзастосунку для автоматизації процесу обробки та створення унікальних екземплярів документів на основі завантаженого документу-шаблону.

Для виконання поставленого завдання необхідно створити блок-схему для відображення послідовності роботи розроблюваного ПП. Алгоритм виконання програми наведено на рис. 3.1.

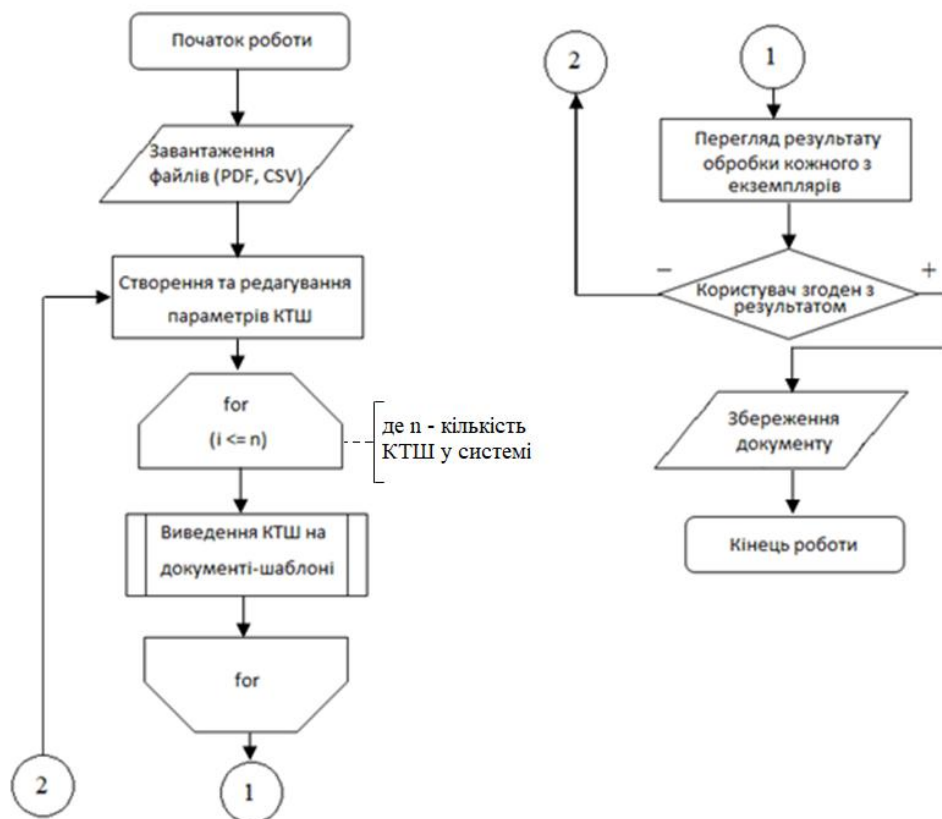


Рисунок 3.1 – Блок-схема загального алгоритму роботи програми

Вхідними даними до ІС є документ формату PDF в якості документа-шаблону і текстовий CSV документ за даними, що повинні бути занесені до нові екземпляри документа.

Вхідні дані поділяються на дві окремі групи, кожна з яких має власний алгоритм обробки та внесення оброблених даних до модуля пам'яті ВЗ.

Функціонування програми забезпечується шляхом послідовного зчитування даних про КТШ, що були введені користувачем, та їх подальшого виведення поверх документу-шаблону, завдяки чому користувач може представити як повинен виглядати кожен з нових екземплярів, що будуть створені за допомогою розробленого ПП.

3.1 Алгоритм вводу даних файлів до системи

Для початку роботи з ПП необхідно завантажити файли, що є обов'язкові для організації процесу генерування екземплярів документів.

Модуль вводу-виводу даних виконує функцію завантаження файлів до ІС, їх інтерпретації у дані, зрозумілі системі, та зберігаються у модулі збереження даних. Модуль вводу-виводу підтримує завантаження файлів двох форматів: pdf та csv. Для кожного з цих форматів виконується свій процес інтерпретації даних.

Файли формату CSV обробляються за допомогою інструменту «рараParse». Цей інструмент спеціалізується на обробці текстових CSV файлів та перетворення їх вмісту до типів та структур даних, зрозумілій мові JavaScript. Для виконання процесу обробки йому необхідно знати який файл треба обробити, при необхідності надати додаткові параметри, серед яких тип кодування тексту файлу, символ розподілення тексту, чи необхідно зберегти назви параметрів даних та їх використання для збереження у зрозумілому формату як для розробника, так і для системи [19].

В даному ПП ми використовуємо даний інструмент для створення структури даних у вигляді масиву, що містить у собі об'єкти. Усі об'єкти мають однаковий набір параметрів, що робить таку структуру можливою до ітераційного перебору даних з опором на стійку структуру даних. Завдяки такій структурі модуль обробки має доступ до списку імен параметрів об'єктів, за допомогою яких може виконуватись як процес ідентифікації динамічного фрагменту, так і визначення інформації для заміщення на місці динамічного фрагменту.

Файли формату PDF після завантаження до ІС одразу ж передаються до модуля зберігання даних, в якому файл зберігається у завантаженому вигляді. Сам процес обробки PDF документів виконується за допомогою бібліотеки «React-PDF». Цей інструмент спеціалізується на створенні області відображення вмісту PDF документів з наданням можливості навігації, збільшення та зменшення масштабу перегляду вмісту документа. Також ця бібліотека надає можливість відстежувати процес завантаження файлу, надавати повідомлення про помилки, відсутність файлу або неможливість відобразити сторінку документа [16].

Для функціонування даної бібліотеки необхідно мати з'єднання з мережею Інтернет, так як процес обробки PDF документів виконується віддаленим сервером. В разі відсутності Інтернет з'єднання бібліотека повертає повідомлення про відсутність з'єднання з віддаленим сервером.

3.2 Використання синтаксичного методу модулем обробки даних

Для генерування унікального екземпляра документа та його виведення з ІС до пам'яті комп'ютера було вирішено використати алгоритм обробки тексту з використанням синтаксичного методу, що є більш гнучким для розширення можливостей алгоритму та оптимальним для обробки великої кількості екземплярів за короткий відрізок часу. Але перед реалізацією даного алгоритму необхідно пройти декілька етапів.

Не першому етапі необхідно створити алгоритм визначення координат розміщення КТШ на площині, що відображає дані завантаженого документа, для чіткого розміщення фрагменту, що відображатиметься на документі. Інформація про координати розміщення, текстові фрагменти, розмір та колір тексту використовується модулем обробки даних.

Модуль обробки даних виконує функцію аналізу інформації про кожний КТШ, що були передані до нього від модуля зберігання даних через центральний модуль. В процесі аналізу використовується алгоритм, за допомогою якого для кожного КТШ виконується процес його відображення на документу-шаблоні.

Параметри КТШ описані в табл. 3.1.

Таблиця 3.1 – Параметри КТШ

Параметр	Характеристика
Текст	Містить інформацію щодо текстових даних, які підлягають аналізу та редагуванню модулем обробки.
X	Вказує на положення компоненту на документі по горизонталі у системі координат, початок якої знаходиться у верхньому-лівому кутку документа.
Y	Вказує на положення компоненту на документі по горизонталі у системі координат, початок якої знаходиться у верхньому-лівому кутку документа.
Розмір	Вказує на розмір тексту, що буде виведений на документі-шаблоні.
Міжстрокова відстань	Вказує на відстань між строками тексту, що використовується в разі редагування тексту у декілька рядків.
Колір	Вказує на колір тексту, що буде виведений на документі-шаблоні.

Після обробки КТШ користувач має можливість переглянути результат використання створеного ТШ з використанням будь-якого запису с текстового документа, що був завантажений до системи. Якщо результат задовольняє вимогам, користувач має можливість завантажити документ на

пам'ять комп'ютера, кожна сторінка якого містить окремий екземпляр документу.

Для перегляду кожного з екземплярів та збереження документів окрім алгоритму розміщення також використовується синтаксичний метод аналізу тексту, що міститься у параметру КТШ «Текст».

Принцип виконання синтаксичного алгоритму полягає у пошуку та заміщенню динамічних фрагментів тексту на відповідні дані з текстового документу. Динамічний фрагмент тексту має синтаксис « $\{\text{ім'я_параметру}\}$ », де під ім'ям параметру мається на увазі назва параметру запису з текстового документу. Це надає можливість використовувати даний алгоритм під будь-які структури вхідних даних. У разі, якщо алгоритм не знайшов параметру, ім'я якого повинно відповідати тексту у фігурних скобках, алгоритм ігнорує цей фрагмент та продовжує процес аналізу.

Синтаксичний алгоритм легкий для розширення можливостей редагування тексту. Наприклад, для перенесення тексту на наступний рядок у відповідному місті, замість перенесення використовується набір символів « $\backslash n$ ». В результаті його використання ми отримаємо текст, що складається з двох рядків та розподілений у місці використання цього синтаксису.

3.3 Процес повної генерації та збереження документу

Для формування вихідного документу, що є результатом виконання ПП, застосовується бібліотека «jsPDF». Ця бібліотека спеціалізується на завданнях створення, редагування та зберігання PDF документів у пам'яті комп'ютера. Має доволі широкий асортимент інструментів для візуалізації усіх відомих форм відображення даних, що робить цю бібліотеку ідеальним для використання інструментом у будь-якому ВЗ, метою якого є реалізація процесу обробки, редагування та створення PDF документів [17].

Для організації процесу збереження документу у пам'яті комп'ютера необхідно організувати покроковий алгоритм виконання роботи модуля обробки та алгоритму роботи бібліотеки «jsPDF». Перед процесом генерування екземплярів необхідно створити об'єкт класу «jsPDF», необхідним для накопичення інформації про екземпляри та його подальшого опрацювання бібліотекою. Надалі виконується процес друку ТШ на документі-шаблоні з подальшим аналізом тексту кожного з КТШ за допомогою синтаксичного методу. Після завершення процесу обробки йде процес транспортування отриманого екземпляру до об'єкту накопичувача та збереження отриманих даних на окремій сторінці. Цей процес виконується для кожного запису, отриманого з текстового CSV документу.

Після завершення процесу обробки всіх екземплярів виконуватиметься процес компіляції накопичених даних бібліотекою у новий PDF документ та його збереження у пам'яті комп'ютера. Файл за замовчуванням зберігається у системній папці «Загрузки» або «Downloads» (у залежності від основної мови операційної системи) з назвою «defaultCertificateName.pdf».

3.4 Висновок

В даному розділі було розроблено схему алгоритму виконання програми, розглянуто та описано основні рішення та процеси, що були створені для вирішення задачі генерування сертифікатів та документів.

Було описано всі етапи існування даних з моменту їх вводу до ІС до моменту їх виведення у форматі документу, що містить у собі унікальні екземпляри.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення й умови використання

Розроблений ПП призначений для вирішення проблеми автоматизації процесу генерування сертифікатів за допомогою використання синтаксичного алгоритму обробки тексту.

Для роботи з розробленим ВЗ система повинна відповідати мінімальним системним вимогам:

- операційна система Microsoft Windows 10/8/7, Linux;
- двоядерний центральний процесор з тактовою частотою 1,8 ГГц;
- 4 ГБ оперативної пам'яті;
- 1 ГБ вільного місця на жорсткому диску
- доступ до мережі Інтернет.

4.2 Характеристики програмного продукту

Розроблюваний ПП складається з наступних модулів:

- центральний модуль. Цей модуль відповідає за виведення графічного інтерфейсу та встановлення зв'язку і взаємодії інших модулів ІС. Був створений та реалізований за допомогою бібліотеки ReactJS, що надає можливість створювати та налагоджувати архітектуру проекту та створювати для організації користувацьких компонентів;

- зберігання даних. Для організації серед зберігання даних використовується бібліотека «React-Redux». Для створення спеціальної області зберігання параметрів КТІШ було використано бібліотеку «Redux-Form»;

- вводу-виводу даних. Цей модуль відповідає за процес завантаження, обробки та передачі отриманих даних з файлів до модуля зберігання даних. Даний модуль надає можливість завантажувати дані з файлів PDF та CSV, кожен з яких проходить власний алгоритм трансформації

у зрозумілі для ІС структури даних. Обробка файлів формату CSV виконується за допомогою інструмента «papaParse», що спеціалізується на виконанні завдань аналізу, обробки, форматування та виведення вмісту файлу до ІС. Файли формату PDF завантажуються та зберігаються у модулі зберігання даних у вигляді спеціального об'єкту File, інформація якого передається до центрального модуля для подальшого виведення вмісту документу;

– редагування. Цей модуль використовується користувачем для створення та зміну параметрів КТШ. Даний модуль реалізований за допомогою налагодження зв'язку зі спеціальною областю збереження інформації про КТШ, що організована за допомогою бібліотеки «Redux-Form», при взаємодії з яким виконується процес отримання та редагування даних з форми;

– обробки. Цей модуль виконує аналіз та перетворення ТШ та документу-шаблону в унікальні екземпляри, що відображають дані з текстового CSV документу. В цьому модулі було реалізовано алгоритм, що зчитує інформацію про КТШ з модуля зберігання даних для їх відображення на документі шаблоні. Також було реалізовано використання синтаксичного методу, за допомогою якого виконується процес аналізу та підстановки динамічних фрагментів тексту КТШ даними з записів CSV документу і створюється і подальшого оформлення унікальних екземплярів документа.

4.3 Звернення до програмного продукту

Для звернення до актуальної версії розроблюваного ВЗ необхідно у браузері відкрити сторінку «<http://certificate-generator.herokuapp.com>», що зберігається на віддаленому сервері.

Для звернення до розроблюваної версії ВЗ необхідно мати налагоджений інструмент для взаємодії з сервісом GitHub та скопіювати у необхідну директорію репозиторій з сервера зазначеного сервісу. Після

цього необхідно викликати системну консоль у контексті локальної директорії, що була клонована, встановити усі пакети бібліотеки та інструментів за допомогою команди «npm install». Після успішного встановлення всіх пакетів необхідно запустити ПП за допомогою команди «npm run dev», що починає виконувати ПП у режимі розробки та виконує компіляцію проекту після кожної збереженої зміни у фалах проекту. Для відкриття ВЗ у локальній мережі необхідно у браузері відкрити сторінку за адресою «http://localhost:3000», за якою надається доступ до розроблюваної версії ВЗ.

4.4 Вхідні та вихідні дані

В якості вхідних даних розроблюваного ПП використовуються текстові файли формату CSV. На рис. 4.1 наведено приклад вмісту текстового CSV документу, що містить інформацію про учасників міжнародного проекту.

	A	B	C	D
1	#	Name	University	Event
2	1	Maxim Andrieiev	National University "Zaporizhzhia Polytechnic"	Winter School 2021
3	2	Olga Maximova	National University "Zaporizhzhia Polytechnic"	Winter School 2021
4	3	Anya Stolyarenko	National University "Zaporizhzhia Polytechnic"	Winter School 2021
5	4	Ivan Ivanov	National University "Zaporizhzhia Polytechnic"	Winter School 2021
6	5	Oleg Voronnoy	National University "Zaporizhzhia Polytechnic"	Winter School 2021
7	6	Katya Lyaschenko	National University "Zaporizhzhia Polytechnic"	Winter School 2021
8	7	Egor Samoylov	National University "Zaporizhzhia Polytechnic"	Winter School 2021
9	8	Nicholas Winters	National University "Zaporizhzhia Polytechnic"	Winter School 2021
10	9	Steve Rogers	National University "Zaporizhzhia Polytechnic"	Winter School 2021
11	10	Katya Lyaschenko	National University "Zaporizhzhia Polytechnic"	Winter School 2021

Рисунок 4.1 – Інформація про учасників міжнародного проекту

Як видно з рис. 4.1 у першому стовпчику вказано номер учасника, у другому – ім'я та прізвище, у третьому – місце проведення, у четвертому – назву проекту або події, в якому учасники приймали участь.

Також в якості вхідних даних необхідним є завантаження документу-шаблону у форматі PDF. На рис. 4.2 зображено приклад документу-шаблону, що міститься у документах формату PDF.

Fachhochschule Dortmund
University of Applied Sciences and Arts

CERTIFICATE

This certificate is awarded to:

NAME _____

FROM _____

For support in organisation of

Prof. Dr. Stephan Recker, Fachhochschule Dortmund

Prof. Galyna Tabunshchik, National University "Zaporizhzhia Polytechnic"

Prof. Sergiy Bushuyev, Kyiv National University of Construction and Architecture

Prof. Anatoliy Sachenko, Ternopil National Economic University

CERTIFICATE

Рисунок 4.2 – Інформація про документ-шаблон

Згідно з рис. 4.2 можна побачити у якому вигляді розроблюваний ПП відображає інформацію після введення PDF документу до ІС.

У якості вихідних даних виступає новий PDF документ, кожна сторінка якого містить унікальні екземпляри, що були згенеровані в результаті виконання алгоритмів модулем обробки. На рис. 4.3 відображено сторінку-екземпляр створеного документу.



Рисунок 4.3 – Екземпляр створеного документу

4.5 Повідомлення

Під час роботи розроблений ПП генерує повідомлення, що описані в табл. 4.1.

Таблиця 4.1 – Повідомлення під час роботи ВЗ

Повідомлення	Опис
Обраний файл не відповідає формату "PDF"	Повідомлення означає, що формат завантаженого файл не відповідає очікуваному формату PDF.
Обраний файл не відповідає формату "CSV"	Повідомлення означає, що формат завантаженого файл не відповідає очікуваному формату CSV.

4.6 Висновок

В даному розділі було проведено опис призначення та визначено умови використання розробленого ВЗ, визначено мінімальні системні вимоги для функціонування ПП. Було надано інструкції, за якими визначається можливості доступу до ЗВ. Було описано характеристики ПП, в яких визначено модулі та їх функції. Описано характеристику та надано приклади вхідних та вихідних даних, надано перелік повідомлень для користувача, за допомогою яких ПП взаємодія з користувачем у випадку виникнення помилок у процесі виконання.

5 КЕРІВНИЦТВО ОПЕРАТОРА

5.1 Призначення й умови виконання програмного продукту

Розроблений ПП призначений для вирішення проблеми автоматизації процесу генерування сертифікатів за допомогою використання синтаксичного алгоритму обробки тексту.

Для роботи з розробленим ВЗ система повинна відповідати мінімальним системним вимогам:

- операційна система Microsoft Windows 10/8/7, Linux;
- двоядерний центральний процесор з тактовою частотою 1,8 ГГц;
- 4 ГБ оперативної пам'яті;
- 1 ГБ вільного місця на жорсткому диску
- доступ до мережі Інтернет.

5.2 Виконання програмного продукту

Після того, як сторінка ВЗ була завантажена, буде відображено головний інтерфейс. На рис. 5.1 зображено головний інтерфейс ПП.



Рисунок 5.1 – Головний інтерфейс ВЗ

Для завантаження документу-шаблону до ВЗ необхідно натиснути кнопку «Завантажити PDF файл», що знаходиться у верхньому-правому кутку інтерфейсу, після чого необхідно обрати файл з пам'яті комп'ютера для завантаження до ІС. Обов'язковою умовою для файлу, що завантажується, є відповідність його формату до PDF. У разі невідповідності форматів буде виведено повідомлення «Обраний файл не відповідає формату "PDF"».

Після завантаження PDF документу файл буде оброблено та виведено в області перегляду документу-шаблону. На рис. 5.2 зображено область головного інтерфейсу, на якому відображено зміст завантаженого документу.

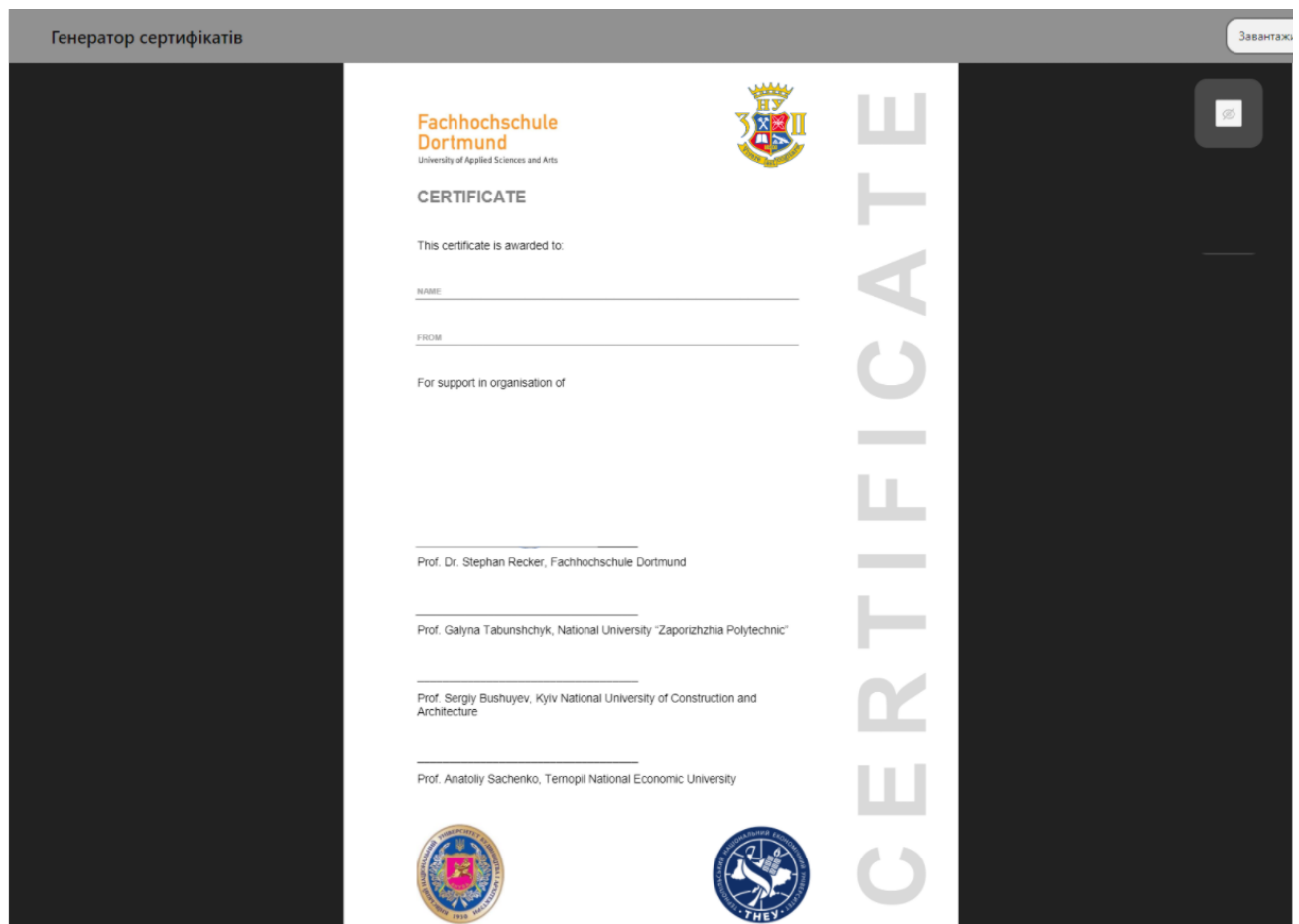


Рисунок 5.2 – Відображення вмісту завантаженого PDF документу

Далі необхідно завантажити текстовий CSV документа до ІС. Для цього необхідно натиснути на кнопку «Завантажити CSV файл», що

знаходиться у верхньому-правому кутку інтерфейсу, після чого необхідно обрати CSV файл з пам'яті комп'ютера. Файл обов'язково повинен мати формат даних CSV. У разі невідповідності форматів буде виведено повідомлення «Обраний файл не відповідає формату "CSV"».

Після завантаження CSV документу файл буде оброблено та виведено в боковій панелі у розділі «Завантажений список», що відображає оброблений вміст документу. На рис. 5.3 зображено вміст текстового документу у вигляді списку.

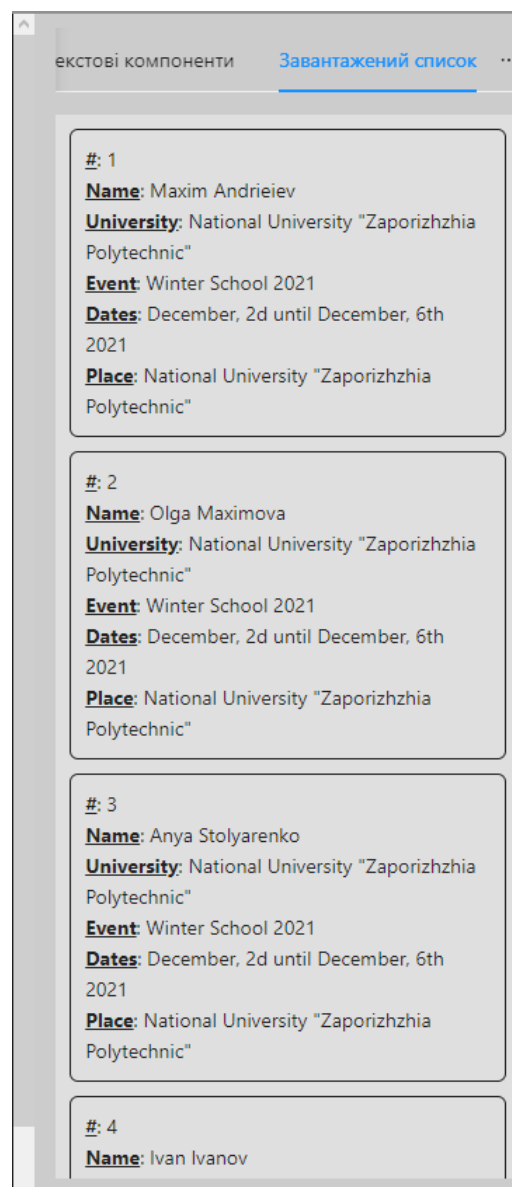


Рисунок 5.3 – Відображення CSV даних у вигляді списку

Для початку створення ТШ необхідно створити його перший компонент. Для створення першого КТШ необхідно у боковій панелі відкрити розділ «Текстові компоненти» та натиснути на кнопку «Створити текстовий компонент», після чого буде відображено список компонентів. Кожен елемент списку являє собою окрему форму, за допомогою якої виконується контроль параметрів КТШ. На рис. 5.4 відображено інтерфейс форми контролю параметрів КТШ.

The image shows a web-based form for creating a text component. At the top, there are two tabs: 'Текстові компоненти' (Text components) and 'Завантажений список' (Loaded list). Below the tabs is a button labeled 'Створити текстовий компонент' (Create text component). The form itself is a modal window with a title bar containing three icons: an information icon, a question mark, and a close button. The form contains several input fields: a text area for 'Текст' (Text), a text box for 'X', a text box for 'Y', a text box for 'Розмір' (Size), a text box for 'Міжстрокова відстань' (Line spacing), and a color picker for 'Колір' (Color).

Рисунок 5.4 – Інтерфейс форми контролю параметрів КТШ

У даній формі існують наступні елементи:

- іконки з підказками щодо додаткового синтаксису та доступних тегів, що можна використовувати для створення динамічного фрагменту ТШ
- елементи вводу тексту КТШ та його параметрів
- кнопка видалення КТШ зі списку

Детальний опис елементів форми КТШ наведено у таблиці 5.1.

Таблиця 5.1 – Опис елементів форми КТШ

Елемент	Опис
Поле вводу «Текст»	Відповідає за текст, що буде виведений на документі. Містить у собі статичний та динамічний фрагменти тексту. Доступні для використання фрагменти виведені у підказці, що з'являється при наведенні на іконку «?».
Поле вводу «Х»	Відповідає за положення КТШ на площині по горизонталі, починаючи від верхнього-лівого кута.
Поле вводу «У»	Відповідає за положення КТШ на площині по вертикалі, починаючи від верхнього-лівого кута.
Поле вводу «Розмір»	Відповідає за розмір тексту, що буде відображений на документі-шаблоні.
Поле вводу «Міжстрокова відстань»	Відповідає за відстань між рядками тексту у випадку, якщо КТШ містить у собі два чи більше рядків. Для перенесення частини тексту на наступний рядок необхідно написати комбінацію символів «\n», що зазначено у підказці, що з'являється при наведенні на іконку «!».
Поле вводу «Колір»	Відповідає за визначення кольору, яке буде застосовано при виведенні тексту на шаблоні.
Іконка «і»	Надає додаткову інформацію про синтаксис, що може бути застосований для організації відображення тексту.
Іконка «?»	Надає додаткову інформацію про доступні значення динамічних фрагментів. Усі фрагменти, що не відповідають зазначеним у підказці, будуть вважатись статичним текстом.
Кнопка закриття	Надає можливість видалити КТШ з пам'яті програми та його виведення на документі-шаблоні.

В області виведення документу-шаблону буде відображений текст КТШ у відповідності з даними, що були введені у формі.

Приклад заповненої форми КТШ відображено на рис. 5.5.

Результат виведення КТШ в області відображення документу-шаблону показано на рис. 5.6.

Рисунок 5.5 – Приклад заповненої форми КТШ

Рисунок 5.6 – Виведення КТШ в області відображення документу-шаблону

Після виконання процесу створення та оформлення ТШ, у відповідності до прикладу, можна переглянути результат роботи у вікні перегляду та завантаження. В ньому буде відображено область виведення документу-шаблону з проаналізованим та обробленим ТШ, де буде відображено результат генерування екземпляру.

Для відкриття вікна перегляду результату обробки необхідно натиснути на кнопку з відображенням ока, що знаходиться у верхньому-правому кутку області виведення кожного екземпляру документу шаблону. Для вибору екземпляра, що буде виведений в інтерфейсі, необхідно натиснути на елемент з бокової панелі вікна, що містить список об'єктів із завантаженого CSV документу. Інтерфейс вікна відображення результату наведено на рис. 5.7.

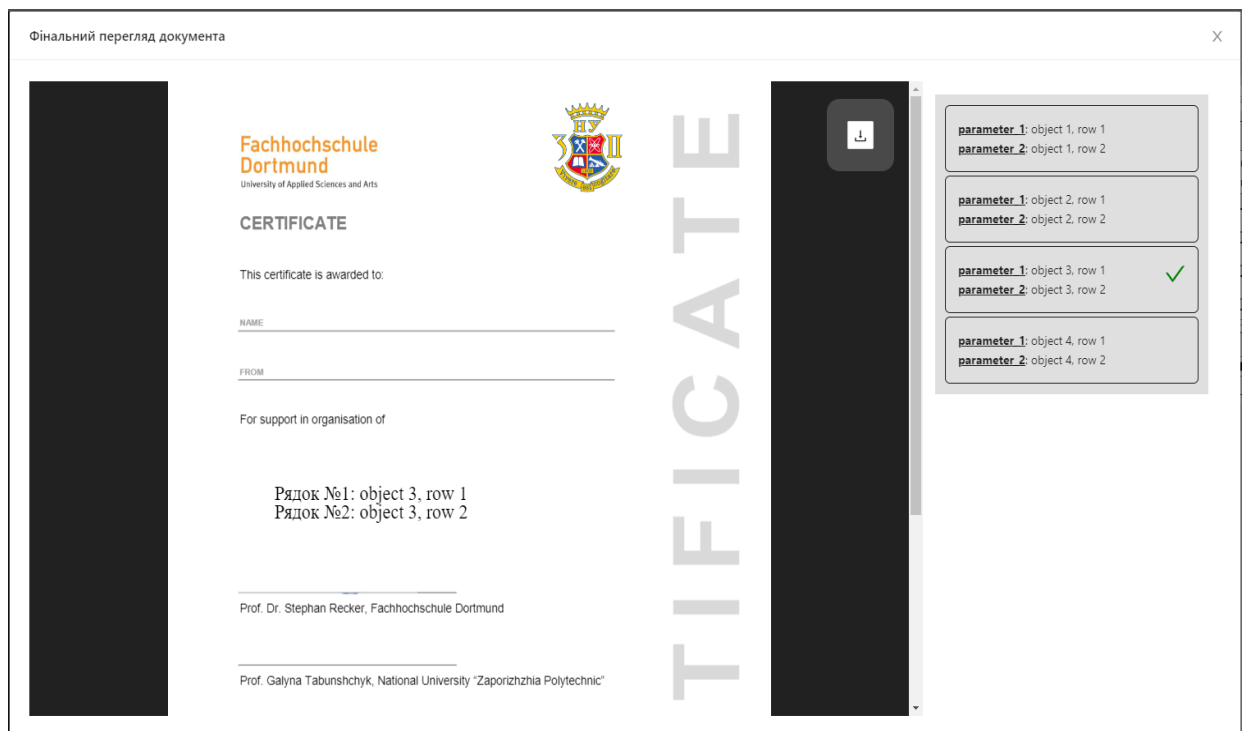


Рисунок 5.7 – Інтерфейс вікна відображення створеного екземпляру

Після перегляду результату генерування екземпляру користувач має можливість завантажити вихідний документ. Документ буде містити по

одному створеному екземпляру на окремій сторінці. Для завантаження вихідного документу необхідно натиснути кнопку завантаження, що знаходиться у верхньому-правому кутку області перегляду екземплярів документу. Після цього буде збережено новий PDF документ з назвою «defaultCertificateName.pdf», що буде збережений у системній папці «Загрузки».

5.3 Повідомлення оператору

Під час роботи розробленого ПП відображаються повідомлення, які призначені для інформування користувача про помилки. Інформація щодо можливих повідомлень та опис їхнього змісту наведені у таблиці 4.1 в розділі «Керівництво програміста».

5.4 Висновок

В даному розділі були наведені необхідні апаратні вимоги для функціонування програми та інструкцію організації процесу роботи з нею. Також описано усі доступні розділи інтерфейсу, їх компоненти та принципи їх роботи. Було описано процес створення КТШ та наведено приклади їх редагування користувачем, пояснено логіку написання тексту для їх розподілу на статичні та динамічні фрагменти для його подальшого використання у процесі генерування екземплярів документу та його збереження у пам'яті комп'ютера. Також було зазначено вимоги до вхідних даних та наведено приклади даних, що вони містять, та охарактеризовано формат вихідних даних програми.

ВИСНОВКИ

Під час виконання роботи було виконано дослідження предметної області та вивчено проблематику поставленої задачі, доведено актуальність її вирішення. В даній роботі була розглянута проблема автоматизації процесу генерування сертифікатів та інших типів документів. Що являє собою комплекс алгоритмів, необхідних для організації процесу створення текстових шаблонів та використання синтаксичного методу для аналізу та обробки тексту компонентів шаблону.

Було проведено порівняльний аналіз існуючих методів створення документів, що використовуються для генерування однотипних документів з унікальними ключовими даними та обрано найбільш оптимальний для вирішення поставленого завдання.

Метою роботи є програмна реалізація використання синтаксичного методу обробки у комбінації з спеціальними алгоритмами обробки текстового шаблону.

Для виконання поставленої мети було виконано наступні завдання:

- створено метод для завантаження та відображення документу в ІС;
- створено метод для завантаження та перегляду даних текстового CSV файлу;
- створено метод для створення та організації ТШ для завантаженого документу;
- створено метод відображення ТКШ в області відображення завантаженого документу;
- створено метод обробки тексту КТШ, з використанням синтаксичного методу обробки;
- створено метод відображення кожного із згенерованих екземплярів документа;
- створено метод завантаження усіх згенерованих екземплярів документа до пам'яті комп'ютера.

Також було проведено дослідження щодо мов програмування та середовища розробки, що використовуються для розробки вебзастосунків. Було проведено аналіз мови програмування, вказано переваги та недоліки кожної з них та створено порівняльну характеристику. В ході виконання роботи було обрано мову програмування JavaScript, тому що вона підтримується більшістю браузерів, надає широкий асортимент інструментів для вирішення дуже широкого спектру завдань, зокрема методів роботи з CSV та PDF файлами та працює доволі швидко у порівнянні з іншими мовами для розробки вебзастосунків. В якості середовища розробки було обрано Visual Studio Code, так як його можна легко налагоджувати під власні потреби розробника, має інструмент авто доповнення коду з використанням технології IntelliSense, знаходиться у вільному доступі та постійно оновлюється розробниками.

Для роботи було розроблено структурну схему розробленого вебзастосунку. Для зберігання даних інформаційної системи було розроблено відповідний модуль з використанням бібліотеки React-Redux.

Графічний інтерфейс та загальну структуру вебзастосунку було розроблено за допомогою бібліотеки ReactJS.

В результаті роботи було розроблено програмну систему для автоматизації процесу генерування сертифікатів міжнародних проектів шляхом розробки алгоритму створення текстового шаблону та використання синтаксичного методу для аналізу та перетворення текстового шаблону в окремий екземпляр.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Руководство к Своду знаний по управлению проектами (Руководство PMBOK) — Пятое издание. — Project Management Institute, Inc., 2013.
2. СЕРТИФІКАТ //Юридична енциклопедія : [у 6 т.] / ред. кол. Ю. С. Шемшученко (відп. ред.) [та ін.]. — К. : Українська енциклопедія ім. М. П. Бажана, 1998—2004. — Т. 5 : П — С.
3. Грамота // Юридична енциклопедія : [у 6 т.] / ред. кол. Ю. С. Шемшученко (відп. ред.) [та ін.]. — К. : Українська енциклопедія ім. М. П. Бажана, 1998-2004. — Т. 1 : А — Г.
4. Грамота // Словник української мови : в 11 т. — К. : Наукова думка, 1970—1980.
5. Електронні документи // Юридична енциклопедія : [у 6 т.] / ред. кол. Ю. С. Шемшученко (відп. ред.) [та ін.]. — К. : Українська енциклопедія ім. М. П. Бажана, 1998. — Т. 2 : Д — Й. — 744 с.
6. PDF24 Tools: Free and easy to use online PDF Tools [Electronic Resource] – Access mode: <https://tools.pdf24.org>
7. Onlnie PDF Converter - Edit, rotate and compress[Electronic Resource] - Access mode: <https://www.pdf2go.com>
8. Microsoft Office - free online Microsoft Office Tools [Electronic Resource] - Access mode: <https://https://www.office.com>
9. JavaScript | MDN [Electronic Resource] – Access mode: <https://developer.mozilla.org/ru/docs/Web/JavaScript>
10. Welcome to Python.org [Electronic Resource] – Access mode: <https://www.python.org>
11. PHP: Hypertext Preprocessor [Electronic Resource] – Access mode: <https://www.php.net>
12. WebStrom: The smartest JavaScript IDE by JetBrains [Electronic Resource] – Access mode: <https://www.jetbrains.com/ru-ru/webstorm/>

13. Visual Studio Code – Code Editing. Redefined [Electronic Resource]
– Access mode: <https://code.visualstudio.com>
14. React – JavaScript бібліотека для створення користувацьких інтерфейсів [Electronic Resource] – Access mode: <https://uk.reactjs.org>
15. React Redux: Official React bindings for Redux [Electronics Resource] – Access mode: <https://react-redux.js.org>
16. React-pdf [Electronic Resource] – Access mode: <https://react-pdf.org>
17. jsPDF – Client-side JavaScript PDF generation for everyone [Electronics Resource] – Access mode: <https://github.com/MrRio/jsPDF>
18. Redux Form – The best way to manage your form state in Redux [Electronics Resource] – Access mode: <https://redux-form.com>
19. Papa Parse – Powerful CSV Parser for Javascript [Electronic Resource] – Access mode: <https://www.papaparse.com>

ДОДАТОК А
Технічне завдання

Вступ

В результаті виконання роботи була досліджена проблема автоматизації процесу генерування унікальних екземплярів документів, яку біло вирішено шляхом розробки алгоритму обробки з використанням синтаксичного методу обробки тексту.

ПП «Certificate Generator» має можливість завантажувати файли, зберігати їх дані, визначати структуру даних для її використання у синтаксичному алгоритмі для генерування екземплярів документів, створювати ТШ за шляхом створення та налагодження усіх його компонентів та зберігати створені екземпляри у пам'яті комп'ютера у вигляді PDF файлу.

A.1 Мета розробки і її призначення

Об'єкт дослідження – процес автоматичної генерації документів.

Предмет дослідження – вебзастосування для генерування сертифікатів.

Мета роботи – покращення адміністративних процесів керування міжнародними проєктами за рахунок генерування відповідних грамот та сертифікатів у форматі PDF документів.

Актуальність полягає в тому, що більшість міжнародних проєктів та організацій зустрічаються з проблемою створення та заповнення великої кількості екземплярів сертифікатів для всіх учасників програм та подій. Чим більша кількість учасників, тим більше часу виділяється для створення та заповнення сертифікатів власноруч. Це призвело до необхідності повної автоматизації заповнення сертифікатів за короткий термін часу, не зважаючи на кількість учасників.

А.2 Стислий зміст основних теоретичних посилок до розробки на основі проведених пошуків

Під створенням сертифікаційних документів мається на увазі повна розробка документу. Цей процес включає в себе створення дизайну (підбір розміру документу, основного фону, базових кольорів, створення рамки, розміщення емблем), оформлення тексту (підбір розміру тексту, стилю, кольору), розміщення областей та їх заповнення для тексту з інформацією кожного з учасників та подальший друк кожного з екземплярів.

Заповнення сертифікатів, за першого погляду, здається доволі простим процесом, що не повинен виконуватися протягом тривалого часу. Насправді час, що витрачається на заповнення всіх документів, прямо пропорційно залежить від кількості учасників, що приймали участь у проектах, подіях або змаганнях.

А.3 Підстави для розробки

Підставою для виконання розробки є завдання на дипломний проєкт на тему “Розроблення вебзастосунку для генерування сертифікатів для учасників міжнародних проєктів”, затверджене наказом по НУ «Запорізька політехніка» № 103 від 30 березня 2020 року.

А.4 Основні вимоги до програми, що розробляється, початковим даним і результату

А.4.1 Вимоги до функціональних характеристик

Вимоги до функціональних характеристик:

- повинно бути реалізовано метод для завантаження та відображення документу в ІС;

- повинно бути реалізовано метод для завантаження та перегляду даних текстового CSV файлу;
- повинно бути реалізовано метод для створення та організації ТШ для завантаженого документу;
- повинно бути реалізовано метод відображення ТКШ в області відображення завантаженого документу;
- повинно бути реалізовано метод обробки тексту КТШ, з використанням синтаксичного методу обробки;
- повинно бути реалізовано метод відображення кожного із згенерованих екземплярів документа;
- повинно бути реалізовано метод завантаження усіх згенерованих екземплярів документа до пам'яті комп'ютера.

A.4.2 Вимоги до надійності

ПП повинна вміти захищати ПП від спроби завантажити файли, тип яких не відповідає вимогам системи.

ПП повинен нормально функціонувати при безперервній роботі комп'ютера та надійному з'єднанні з мережею Інтернет. У разі виникнення помилки у системі необхідно виконати перезапуск сторінки.

A.4.3 Умови експлуатації

Умови експлуатації:

- програмне забезпечення не потребує обслуговуючого персоналу;
- для програмист продуктом необхідно пройти короткий інструктаж по принципу написання тексту з використання синтаксичних елементів.

A.4.4 Вимоги до складу та параметрів технічних засобів

Вимоги до складу та параметрів технічних засобів:

- операційна система Microsoft Windows 10/8/7, Linux;
- двоядерний центральний процесор з тактовою частотою 1,8 ГГц;
- 4 ГБ оперативної пам'яті;
- 1 ГБ вільного місця на жорсткому диску
- доступ до мережі Інтернет.

A.4.5 Вимоги до маркування та упакування

Програмне забезпечення не потребує додаткових дій при маркуванні та упакуванні.

A.4.6 Вимоги до транспортування та збереження

Не існує вимог для транспортування розробленої версії продукту, так як він розміщений у сервісі зберігання віддалених репозиторіїв GitHub.

A.5 Вимоги до програмної документації

Склад програмної документації повинен включати в себе:

- технічне завдання;
- керівництво програміста;
- керівництво оператора;
- опис програми.

А.6 Основні обмеження на установку і використання програмного продукту

Розроблюваний програмний продукт не потребує інсталяції. Програмний продукт повинен використовуватись на комп'ютерах які задовільняють вимогам та параметрам технічних засобів.

ДОДАТОК Б**Опис програмного продукту**

Б.1 Загальні відомості

В результаті виконання роботи була досліджена проблема автоматизації процесу генерування унікальних екземплярів документів, яку біло вирішено шляхом розробки алгоритму обробки з використанням синтаксичного методу обробки тексту.

ПП «Certificate Generator» має можливість завантажувати файли, зберігати їх дані, визначати структуру даних для її використання у синтаксичному алгоритмі для генерування екземплярів документів, створювати ТШ за шляхом створення та налагодження усіх його компонентів та зберігати створені екземпляри у пам'яті комп'ютера у вигляді PDF файлу.

Для роботи з розробленим ВЗ система повинна відповідати мінімальним системним вимогам:

- операційна система Microsoft Windows 10/8/7, Linux;
 - двоядерний центральний процесор з тактовою частотою 1,8 ГГц;
 - 4 ГБ оперативної пам'яті;
 - 1 ГБ вільного місця на жорсткому диску
- доступ до мережі Інтернет.

Б.2 Функціональне призначення

Вимоги до функціональних характеристик:

- повинно бути реалізовано метод для завантаження та відображення документу в ІС;
- повинно бути реалізовано метод для завантаження та перегляду даних текстового CSV файлу;
- повинно бути реалізовано метод для створення та організації ТШ для завантаженого документу;
- повинно бути реалізовано метод відображення ТКШ в області відображення завантаженого документу;

- повинно бути реалізовано метод обробки тексту КТШ, з використанням синтаксичного методу обробки;
- повинно бути реалізовано метод відображення кожного із згенерованих екземплярів документа;
- повинно бути реалізовано метод завантаження усіх згенерованих екземплярів документа до пам'яті комп'ютера.

Б.3 Опис логічної структури

Розроблений ВЗ повинен складатися з наступних модулів:

- центральний модуль;
- зберігання даних;
- вводу-виводу даних;
- редагування;
- обробки.

Розроблений ВЗ використовує дані щодо учасників проектів міжнародних організацій та компаній та PDF-документа, що є трафаретом, необхідним для обробки та подальшого генерування сертифікатів на основі даних учасників в якості вхідних даних. Ці дані можуть бути представлені у наступних форматах:

- текстові файли (CSV);
- універсальні документи (PDF).

Дані формати були обрані для стійкості та надійності даних, що завантажуються, зберігаються та оброблюються для генерування сертифікатів.

В якості центрального модуля було розроблено клієнтський ВЗ з використанням бібліотеки ReactJS. Даний модуль відповідає за відображення збудованого графічного інтерфейсу ПП, виконання сценаріїв та подій програми у результаті взаємодії клієнта з активними елементами інтерфейсу.

Для реалізації модуля зберігання даних була застосована технологія React-Redux. React-Redux являється додатковою службою бібліотеки ReactJS, що надає ПП зв'язок з модулем зберігання даних через взаємодію клієнта з інтерфейсом та проходженням компонентів проекту через життєві цикли існування компоненту. Технологія React-Redux може зберігати усі типи даних, що реалізовані у мові JavaScript, від простих (string, number, symbol, boolean) до складних (object, array). Для взаємодії з кожним елементом модуля зберігання даних необхідно створити методи збереження та видачі конкретних фрагментів пам'яті даного модуля. Це значно спрощує процес створення архітектури проекту та надає можливість чітко контролювати процеси обміну даних.

В якості модуля вводу-виводу даних застосовуються вбудовані до WEB браузера інструменти взаємодії з файловою системою, що необхідні для вводу CSV списку та PDF документа до IC, та збереження обробленого документу у пам'яті пристрою.

Для виводу даних із CSV документів у інтерфейсі ПП використовуються користувацькі компоненти виводу даних, створених та компільованих за допомогою бібліотеки ReactJS.

Для виводу даних щодо завантаженого PDF документу в інтерфейсі ПП було обрано інструмент React-PDF, що спеціалізується на виведенні даних щодо завантажених PDF файлів, яке реалізовано на основі взаємодії інструменту з тегом гіпертекстової розмітки HTML «<canvas>». Це надає можливість малювати та редагувати текстові компоненти та шаблонізувати документ для його подальшої обробки.

Для створення PDF документу перед завантаженням файлу у пам'ять пристрою було обрано бібліотеку jsPDF, що спеціалізується на створенні, редагуванні та збереженні PDF документів.

Для реалізації модуля редагування ТІШ було написано форму, реалізовану за допомогою інструменту Redux-Form, та реалізовано алгоритм розміщення КТІШ у відповідності до даних з форми. Інструмент Redux-Form

надає можливість використовувати окремо виділену область пам'яті, що виділяється за допомогою модуля зберігання даних, для зберігання, редагування та виводу даних без написання методів збереження та отримання введених до форми даних.

При кожній зміні значень будь-якої області пам'яті форми відправляє ПП сигнал про зміну, що дає можливість одразу отримати та вивести нові дані без необхідності сповіщення ІС про зміну. Такий підхід надає можливість спостерігати зміни кожного з КТШ, параметри якого були змінені та одразу ж виведені у інтерфейсі ПП.

Для відображення КТШ в інтерфейсі ІС було написано алгоритм, що має доступ до тегу «<canvas>», котрий містить у собі інформацію про завантажений PDF документ. Алгоритм, перебираючи дані про кожний КТШ, друкував введений текст із параметрами, що були введені у формі редагування КТШ.

Для реалізації модуля обробки було використано алгоритм, що працює по принципу алгоритму, що використовується у модулі редагування ТШ. Але окрім розміщення КТШ на документі, цей алгоритм також проводить аналіз та обробку тексту, що був введений у формі.

Текст, що вводиться до ТКШ, поділяється на два типи:

- статичний;
- динамічний.

Статичний текст є незмінним для кожного екземпляру сертифікату та ігнорується алгоритмом обробки тексту.

Динамічний текст – частина тексту КТШ, що має патерн « $\{\text{текст}\}$ ». За допомогою тексту, що знаходиться у фігурних скобка, виконується пошук значення параметру із CSV документу, завантаженого до ІС. Далі замість даного патерну буде виведено значення з CSV документу.

Після процесу обробки, результат обробки кожного екземпляру передається до модуля виводу даних. За допомогою інструменту jsPDF усі

передані екземпляри передаються до створеного PDF документу, кожна сторінка якого вміщує окремий екземпляр сертифікату.

На рис. 2.1 зображена структурна схема розробленого ПП.

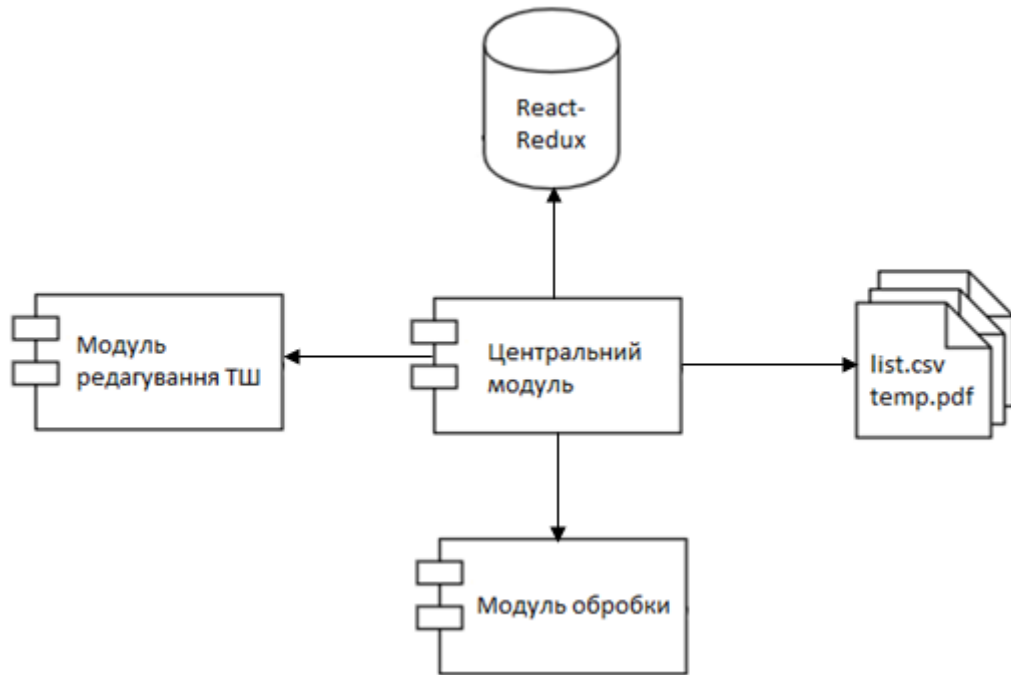


Рисунок Б.1 – Структурна схема програмного продукту

Згідно з рис. Б.1 розроблений ПП використовує дані учасників, що перелічені в CSV файлі, та макет сертифікату у форматі PDF, які опрацьовуються за допомогою спеціальних алгоритмів аналізу КТШ та їх тексту. Інструмент React-Redux було обрано для тимчасового зберігання даних про учасників, та макет сертифікату. За функцію завантаження необхідних даних до ВЗ відповідає модуль вводу-виводу даних, що базується на вбудованих інструментах WEB браузера для його взаємодії з файловою системою. Модуль редагування використовує дані про КТШ, що зберігаються у спеціальному сховищі модуля зберігання даних, та виконує друк тексту поверх макету сертифікату. Для модуля обробки використовується алгоритм аналізу тексту КТШ, результат обробки якого видається модулю вводу-виводу даних для збереження згенерованих екземплярів сертифікатів у

пам'яті комп'ютера. Центральний модуль відповідає за взаємозв'язок усіх вище перерахованих модулів. Даний модуль написаний з використанням бібліотеки React JS, за допомогою якого було розроблено графічний інтерфейс та сценарії взаємозв'язку модулів та користувача з ВЗ.

Б.4 Вибір інструментарію розробки програмного продукту

Для розробки ПП було обрано мову програмування було обрано мову JavaScript, тому що вона підтримується багатьма браузерами, має великий асортимент інструментів та готових рішень для вирішення широкого спектру задач, виконується значно швидше ніж інші мови, що використовуються для розробки ВЗ. Також з його допомогою є можливість створити як клієнтські частини WEB системи, так і серверні програми, що робить обрану мову дуже гнучкою для вирішення будь-якого проектного рішення.

Для розробки ПП було обрано середовище програмування Visual Studio Code, бо воно має можливість налагоджувати середовище, має вбудований помічник для легкого редагування коду та навігації між залежними файлами, постійно модернізується та підтримується не тільки розробниками середовища, але й користувачами, має більший рівень доступності.

Б.5 Виклик і завантаження

Для звернення до актуальної версії розроблюваного ВЗ необхідно у браузері відкрити сторінку «<http://certificate-generator.herokuapp.com>», що зберігається на віддаленому сервері.

Для звернення до розроблюваної версії ВЗ необхідно мати налагоджений інструмент для взаємодії з сервісом GitHub та скопіювати у необхідну директорію репозиторій з сервера зазначеного сервісу. Після цього необхідно викликати системну консоль у контексті локальної

директорії, що була клонована, встановити усі пакети бібліотеки та інструментів за допомогою команди «npm install». Після успішного встановлення всіх пакетів необхідно запустити ПП за допомогою команди «npm run dev», що починає виконувати ПП у режимі розробки та виконує компіляцію проекту після кожної збереженої зміни у фалах проекту. Для відкриття ВЗ у локальній мережі необхідно у браузері відкрити сторінку за адресою «http://localhost:3000», за якою надається доступ до розроблюваної версії ВЗ.

Б.6 Приклади реалізації

Після того, як сторінка ВЗ була завантажена, буде відображено головний інтерфейс. На рис. Б.2 зображено головний інтерфейс ПП.

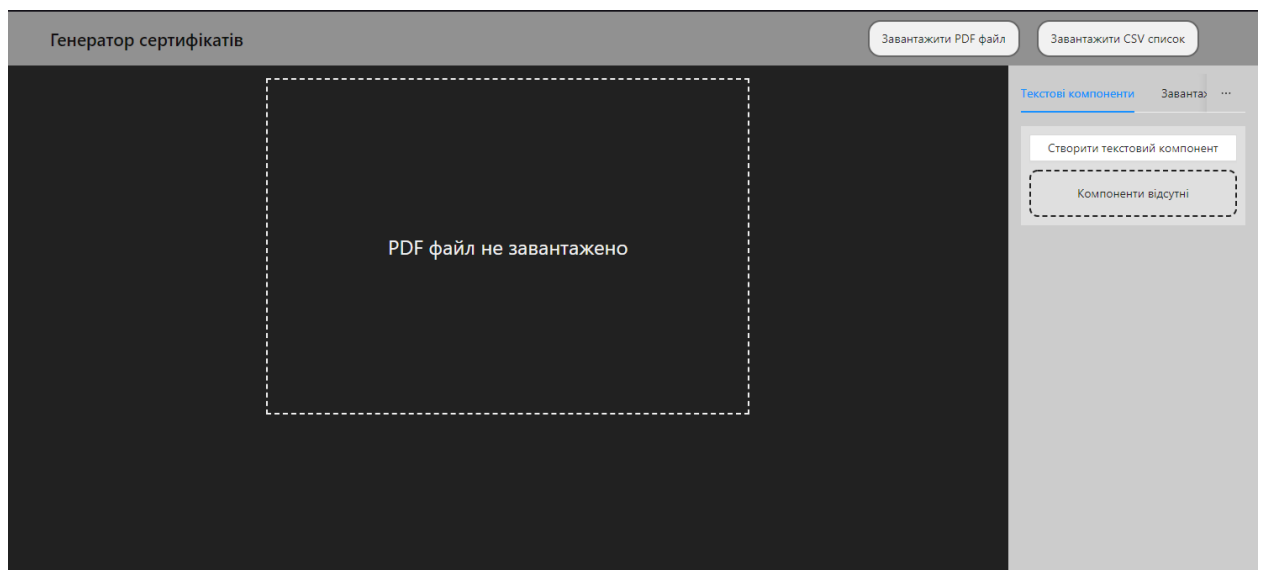


Рисунок Б.2 – Головний інтерфейс ВЗ

Для запуску процесу генерування документів необхідно завантажити документ-шаблон та інформацію про учасників проекту, для чого необхідно натиснути на відповідні кнопки у верхній-правій частині головного інтерфейсу та завантажити відповідні файли. Після цього необхідно створити

та ввести усі необхідні параметри кожного з КТШ у розділі «Текстові компоненти» на боковій панелі для організації ТШ на документу-шаблоні.

Для перегляду усіх екземплярів системи перед завантаженням необхідно натиснути кнопку з символом ока, після чого відкриється вікно відображення оброблених екземплярів у відповідності з ТШ та обраними зі списку даними.

Для збереження згенерованих екземплярів необхідном натиснути на кнопку «Завантажити», що знаходиться у правій-верхній частині області відображення документу у вікні перегляду створених екземплярів, після чого буде проведено процес створення та збереження файл у пам'яті комп'ютера у системній папці «Загрузки» під ім'ям «defaultCertificateName.pdf».

Б.7 Вхідні та вихідні дані

В якості вхідних даних розроблюваного ПП використовуються текстові файли формату CSV. На рис. Б.3 наведено приклад вмісту текстового CSV документу, що містить інформацію про учасників міжнародного проекту.

	A	B	C	D
1	#	Name	University	Event
2	1	Maxim Andrieiev	National University "Zaporizhzhia Polytechnic"	Winter School 2021
3	2	Olga Maximova	National University "Zaporizhzhia Polytechnic"	Winter School 2021
4	3	Anya Stolyarenko	National University "Zaporizhzhia Polytechnic"	Winter School 2021
5	4	Ivan Ivanov	National University "Zaporizhzhia Polytechnic"	Winter School 2021
6	5	Oleg Voronnoy	National University "Zaporizhzhia Polytechnic"	Winter School 2021
7	6	Katya Lyaschenko	National University "Zaporizhzhia Polytechnic"	Winter School 2021
8	7	Egor Samoylov	National University "Zaporizhzhia Polytechnic"	Winter School 2021
9	8	Nicholas Winters	National University "Zaporizhzhia Polytechnic"	Winter School 2021
10	9	Steve Rogers	National University "Zaporizhzhia Polytechnic"	Winter School 2021
11	10	Katya Lyaschenko	National University "Zaporizhzhia Polytechnic"	Winter School 2021

Рисунок Б.3 – Інформація про учасників міжнародного проекту

Як видно з рис. Б.1 у першому стовпчику вказано номер учасника, у другому – ім'я та прізвище, у третьому – місце проведення, у четвертому – назву проекту або події, в якому учасники приймали участь.

Також в якості вхідних даних необхідним є завантаження документу-шаблону у форматі PDF. На рис. Б.2 зображено приклад документу-шаблону, що міститься у документах формату PDF.

Fachhochschule Dortmund
University of Applied Sciences and Arts

CERTIFICATE

This certificate is awarded to:

NAME _____

FROM _____

For support in organisation of

Prof. Dr. Stephan Recker, Fachhochschule Dortmund

Prof. Galyna Tabunshchuk, National University "Zaporizhzhia Polytechnic"

Prof. Sergiy Bushuyev, Kyiv National University of Construction and Architecture

Prof. Anatoliy Sachenko, Ternopil National Economic University

CERTIFICATE

Рисунок Б.4 – Інформація про документ-шаблон

Згідно з рис. Б.4 можна побачити у якому вигляді розроблюваний ПП відображає інформацію після введення PDF документу до ІС.

У якості вихідних даних виступає новий PDF документ, кожна сторінка якого містить унікальні екземпляри, що були згенеровані в результаті виконання алгоритмів модулем обробки. На рис. Б.5 відображено сторінку-екземпляр створеного документу.



Рисунок Б.5 – Екземпляр створеного документу

ДОДАТОК В
Текст програми

В.1 Файл index.tsx

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Provider } from "react-redux";
import { pdfjs } from 'react-pdf';

import './styles/index.css';
import App from './containers/App';
import store from './store';

pdfjs.GlobalWorkerOptions.workerSrc =
`//cdnjs.cloudflare.com/ajax/libs/pdf.js/${pdfjs.version}/pdf.worker.min.js`;

ReactDOM.render(
  <Provider store={store}>
    <App />
  </Provider>,
  document.getElementById('root')
);
```

В.2 Файл index.tsx

```
import React from 'react';
import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';
import App from '../containers/App';

export default (props: any) => (
  <Router>
    <Switch>
      <Route path="/" exact component={App} />
    </Switch>
  </Router>
);
```

В.3 Файл containers/App/index.tsx

```
import React, { useEffect } from 'react';
import { Layout } from 'antd';
```

```

import MainSiderComponent from '../components/App/MainSiderComponent/index';
import MainContentComponent from '../components/App/MainContentComponent/index.jsx';
import MainHeaderComponent from '../components/App/MainHeaderComponent';

const App = () => {
  useEffect(() => {
    document.title = 'Certificate Generator | App';
  }, [])

  return (
    <div className="App">
      <Layout style={{ minHeight: '100vh' }}>
        <MainHeaderComponent />
        <Layout>
          <MainContentComponent />
          <MainSiderComponent />
        </Layout>
      </Layout>
    </div>
  );
}

export default App;

```

В.4 Файл components/App/MainContentComponent/index.tsx

```

import React, { useCallback, useEffect, useRef, useState } from 'react';
import { Layout, Modal, Button, Tooltip } from 'antd';
import { connect } from 'react-redux';
import { Document, Page } from 'react-pdf';

import { getParsedData, getPdfData } from '../../store/reducers/loaded'
import { formValueSelector } from 'redux-form';
import { getPreviewModal, handlePreviewModalContainer } from '../../store/reducers/modalHandlers';
import PreviewModalComponent from '../PreviewModalComponent';
import jsPDF from 'jspdf';
import NoContentUploaded from './noContentUploaded';
import { EyeInvisibleOutlined, EyeOutlined } from '@ant-design/icons';

```

```

const [width, height] = [735, 1040]

const {
  Content
} = Layout

const MainContentComponent = ({
  pdfData,
  parsedData,
  components,
  previewModal,
  handlePreviewModalContainer
}) => {
  const [ isInitialLoad, setIsInitialLoad ] = useState(true)
  const [ imageData, setImageData ] = useState()

  const canvasRef = useRef(null);

  useEffect(() => {
    if (pdfData && canvasRef.current) {
      const ctx = canvasRef.current.getContext('2d');

      if (imageData) {
        const image = new Image()
        image.src = imageData
        ctx.drawImage(image, 0, 0, width, height)
      }

      ctx.fillStyle = "#000000";

      components && Object.values(components).filter(item => !item._is_Deleted_).forEach(({ text, posX,
posY, font, colour, margin }) => {
        if (text && posX >= 0 && posY >= 0) {
          const textLines = text.split("\n");
          if (textLines.length < 2 ) {
            ctx.font = `${font}px serif`
            ctx.fillStyle = colour;
            ctx.fillText(text, posX, posY)
          } else {
            const metrics = ctx.measureText(text)
            let height = metrics.actualBoundingBoxAscent + metrics.actualBoundingBoxDescent;

```

```

      textLines.forEach((Text, i) => {
        ctx.font = `${font}px serif`;
        ctx.fillStyle = colour || '#000000';
        ctx.fillText(Text.trim(), posX, (+posY) + (+height + +(margin || 0)) * i)
      })
    }

  };
})
}

// eslint-disable-next-line react-hooks/exhaustive-deps
}, [pdfData, isInitialLoad, imageData, components]);

const handleImageLoad = useCallback((ref) => {
  if (isInitialLoad) {
    setIsInitialLoad(false)
    setImageData(ref.toDataURL())
  }
}, [isInitialLoad])

const handlePreview = useCallback(() => {
  handlePreviewModalContainer(true)
}, [handlePreviewModalContainer])

const onDocumentLoadSuccess = useCallback(() => {
  setIsInitialLoad(true);
}, [])

const handleDownload = useCallback((canvasRef, drawPage) => {
  if (canvasRef.current) {
    const pdf = new jsPDF();

    parsedData.forEach((item, i) => {
      i && pdf.addPage()
      drawPage(canvasRef, item);
      const image = canvasRef.current.toDataURL('image/jpeg', 1.0)
      pdf.addImage(image, 'JPEG', 0, 0)

    })

    pdf.save('defaultCertificateName.pdf')
  }

```

```

    handlePreviewModalContainer(false)
  }
}, [handlePreviewModalContainer, parsedData])

return (
  <Content style={{ position: 'relative' }}>
    <div className="pdf-container">
      <div className="pdf-container-page">
        <Document
          file={pdfData}
          onLoadSuccess={onDocumentLoadSuccess}
          noData={() => (<NoContentUploaded text="PDF файл не завантажено" />)}
        >
          <Page
            pageNumber={1}
            width={width}
            height={height}
            onRenderSuccess={() => handleImageLoad(canvasRef.current)}
            canvasRef={el => {
              canvasRef.current = el;
            }}
          />
        </Document>

      </div>
    </div>
  </div>
  {
    pdfData ? (
      <div className="pdf-navigator">

        {
          parsedData.length ? (
            <Tooltip title="Перегляд та завантаження">
              <Button onClick={handlePreview} icon={<EyeOutlined />} />
            </Tooltip>
          ) : (
            <Tooltip title="Необхідно завантажити CSV список">
              <Button onClick={handlePreview} icon={<EyeInvisibleOutlined />} disabled />
            </Tooltip>
          )
        }
      </div>
    )
  }

```

```

    <Modal
      title="Фінальний перегляд документа"
      onCancel={() => handlePreviewModalContainer(false)}
      footer={null}
      visible={previewModal}
      forceRender={true}
      okText="Save"
      width={1500}
      centered
    >
      <PreviewModalComponent
        imageData={imageData}
        handleDownload={handleDownload}
      />
    </Modal>
  </div>
): "
}
</Content>
)
}

```

```
const selector = formValueSelector('drawComponents')
```

```

export default connect(
  (state) => ({
    pdfData: getPdfData(state),
    parsedData: getParsedData(state),
    components: selector(state, 'components'),
    previewModal: getPreviewModal(state)
  }),
  (dispatch) => ({
    handlePreviewModalContainer: (val) => dispatch(handlePreviewModalContainer(val))
  })
)(MainContentComponent);

```

B.5 Файл components/App/MainHeaderComponent/index.tsx

```

import React, { useCallback } from 'react';
import papaParse from 'papaparse';
import { Layout } from 'antd';
import { connect } from 'react-redux';

```

```

import Dropzone from 'react-dropzone';

import { saveParsedData, savePdfData } from '../store/reducers/loaded';

const { Header } = Layout

type IDefaultProps = {
  savePdfData: Function,
  saveParsedData: Function,
}

const MainHeaderComponent = ({
  saveParsedData,
  savePdfData
}: IDefaultProps) => {

  const handleUploadCSV = useCallback((files) => {
    console.log('CSV files ===> ', files)
    const isError = !files[0].path.includes(".csv");
    !isError
    ? papaParse.parse(files[0], {
      header: true,
      error: (...args) => {
        console.log(...args)
      },
      complete: ({ data }) => {
        saveParsedData(data)
      },
      dynamicTyping: true,
      skipEmptyLines: true,
    })
    : alert('Обраний файл не відповідає формату "CSV"')
  }, [saveParsedData])

  const handleUploadPDF = useCallback((files) => {
    console.log('PDF files ===> ', files)
    const isError = !files[0].path.includes(".pdf");
    if (isError) {
      alert('Обраний файл не відповідає формату "PDF"')
      return;
    }
    savePdfData(files[0])
  }, [savePdfData])

```

```
}, [savePdfData])
```

```
return (
  <Header>
    <div className="header-container">
      <h2>
        Генератор сертифікатів
      </h2>
      <div className="header-dragg-container">

        <Dropzone onDrop={handleUploadPDF}>
          ({ { getRootProps, getInputProps, isDragActive } }) => {
            return (
              <div
                {...getRootProps()}
                className={isDragActive ? 'dropzone dropzone--isActive' : 'dropzone'}
              >
                <input {...getInputProps()} />
                <span className="m-0">
                  Завантажити PDF файл
                </span>
              </div>
            )
          }
        </Dropzone>

        <Dropzone onDrop={handleUploadCSV}>
          ({ { getRootProps, getInputProps, isDragActive } }) => {
            return (
              <div
                {...getRootProps()}
                className={isDragActive ? 'dropzone dropzone--isActive' : 'dropzone'}
              >
                <input {...getInputProps()} />
                <span className="m-0">
                  Завантажити CSV список
                </span>
              </div>
            )
          }
        </Dropzone>
      </div>
    </div>
  )
)
```

```

        </div>
      </div>
    </Header>
  )
}

export default connect(
  (state: any) => ({
  }),
  (dispatch: (o: any) => any) => ({
    saveParsedData: (val: Array<any>) => dispatch(saveParsedData(val)),
    savePdfData: (val: any) => dispatch(savePdfData(val)),
  })
)(MainHeaderComponent);

```

B.6 Файл MainSiderComponent/index.tsx

```

import React from 'react';
import { Layout, Tabs } from 'antd';
import { connect } from 'react-redux';

import ComponentEditTab from './ComponentEditTab.jsx';
import ListOutputTab from './ListOutputTab';

const {
  Sider
} = Layout

const {
  TabPane
} = Tabs

const MainSiderComponent = () => {

  return (
    <Sider className="main-page-sider">
      <Tabs defaultActiveKey="1">
        <TabPane tab="Текстові компоненти" key="1">
          <ComponentEditTab/>
        </TabPane>
        <TabPane tab="Завантажений список" key="2">
          <ListOutputTab />

```

```

        </TabPane>
      </Tabs>
    </Sider>
  )
}

export default connect(
  (state) => ({
  }),
  (dispatch: (o: any) => any) => ({
  })
)(MainSiderComponent);

```

В.7 Файл MainSiderComponent/ ListOutputTab.tsx

```

import React from 'react';
import { connect } from 'react-redux';

import { getParsedData } from '../store/reducers/loaded';

type IProps = {
  parsedData: Array<any>
}

const ListOutputTab = ({
  parsedData
}: IProps) => {
  return (
    <div className="loader-content csv-list-container">
      {parsedData.length ?
        parsedData.map(item => (
          <div className="csv-list-item">
            {Object.entries(item)
              .map(([ k, v ]) => (
                <div className="csv-list-item-row"> <strong>{k}</strong>: {v || "None"} </div>
              ))}
          </div>
        )) : (
          <div className="no-data-container">
            CSV даних не знайдено
          </div>
        )}
    </div>
  )
}

```

```

    </div>
  )
}

export default connect(
  (state) => ({
    parsedData: getParsedData(state),
  }),
  (dispatch: (o: any) => any) => ({
  })
)(ListOutputTab);

```

В.8 Файл MainSiderComponent/ ComponentEditTab.jsx

```

import React from 'react';
import { connect } from 'react-redux';
import { Button } from 'antd'

import { createDrawComponent, getDrawComponents, removeDrawComponent } from
'../../store/reducers/drawComponents';
import { getParsedData, saveParsedData } from '../../store/reducers/loaded';
import DrawComponentsList from './drawComponentForm';
import { change } from 'redux-form';

const ComponentEditTab = ({
  drawComponents,
  createDrawComponent,
  removeDrawComponent,
  parsedData,
  updateDrawComponent,
}) => {

  return (
    <div className="loader-content">
      <Button onClick={() => createDrawComponent()} className='mb-2'>
        Створити текстовий компонент
      </Button>
      <div className="draw-components-list">
        <DrawComponentsList
          drawComponents={drawComponents}
          removeDrawComponent={removeDrawComponent}
          parsedData={parsedData}

```

```

        updateDrawComponent={updateDrawComponent}
      />
    </div>
  </div>
)
}

export default connect(
  (state) => ({
    drawComponents: getDrawComponents(state),
    parsedData: getParsedData(state),
  }),
  (dispatch) => ({
    saveParsedData: (val) => dispatch(saveParsedData(val)),
    createDrawComponent: (val) => dispatch(createDrawComponent(val)),
    removeDrawComponent: (id) => dispatch(removeDrawComponent(id)),
    updateDrawComponent: (field, val) => dispatch(change('drawComponents', field, val))
  })
)(ComponentEditTab);

```

B.8 Файл MainSiderComponent/drawComponentForm.jsx

```

import React, { useCallback } from 'react';
import { Row, Col, Button, Tooltip } from 'antd';
import { CloseCircleOutlined, InfoCircleOutlined, QuestionCircleOutlined } from '@ant-design/icons';
import { Field, reduxForm } from 'redux-form';
import { debounce } from 'lodash';

const labelWidth = 8
const inputWidth = 16

const renderTextArea = ({ input, label, meta: { valid, invalid, touched, error } }) => (
  <Row className="mb-2">
    <Col span={labelWidth}>
      {label && (<label>{label}</label>)}
    </Col>
    <Col span={inputWidth}>
      <textarea
        id={input.name}
        {...input}
        placeholder={label}
        valid={touched && valid}

```

```

        invalid={touched && invalid}
        style={{ width: '100%' }}
      />
    </Col>
  </Row>
)

const renderNumberInput = ({ input, label, meta: { valid, invalid, touched, error } }) => (
  <Row className="mb-2">
    <Col span={labelWidth}>
      {label && (<label>{label}</label>)}
    </Col>
    <Col span={inputWidth}>
      <input
        id={input.name}
        {...input}
        placeholder={label}
        valid={touched && valid}
        invalid={touched && invalid}
        type="number"
      />
    </Col>
  </Row>
)

const renderColourInput = ({ input, label, meta: { valid, invalid, touched, error } }) => {
  const fn = debounce(input.onChange, 500)

  const debouncedChange = (e) => fn(e.target.value)

  return (
    <Row className="mb-2">
      <Col span={labelWidth}>
        {label && (<label>{label}</label>)}
      </Col>
      <Col span={inputWidth}>
        <input
          id={input.name}
          {...input}
          onChange={debouncedChange}
          placeholder={label}
          valid={touched && valid}

```

```

        invalid={touched && invalid}
        type="color"
      />
    </Col>
  </Row>
)
}

const DrawComponentsList = ({
  parsedData,
  drawComponents,
  removeDrawComponent,
  updateDrawComponent,
}) => {

  const handleComponentRemove = useCallback((id) => {
    removeDrawComponent(id)
    updateDrawComponent(`components.${id}._is_Deleted_`, true)
  }, [removeDrawComponent, updateDrawComponent])

  return (
    <form>
      {
        drawComponents.length ? drawComponents.map(({ type, id, ...rest }, i) => {
          switch(type) {
            case 'text':
              return (
                <div className="draw-components text-draw-component" key={id}>
                  <div className="item-tools-panel">
                    <Tooltip
                      title={`Для переносу тексту на наступний рядок додайте до тексту "\\n"}`
                      placement="leftTop"
                    >
                      <div className="text-label-tip mr-3">
                        <InfoCircleOutlined />
                      </div>
                    </Tooltip>

                    <Tooltip
                      title={
                        parsedData.length

```

```

      ? `Доступні наступні теги: ${Object.keys(parsedData[0] || {}).map(tag =>
`$${tag.toLowerCase()}`)}.join(', ')`
      : 'Не знайдено доступних тегів'
    }
    placement="leftTop"
  >
    <div className="tags-info-tip mr-3">
      <QuestionCircleOutlined />
    </div>
  </Tooltip>

  <Button
    type="primary"
    danger
    className="delete-item-button"
    onClick={() => handleComponentRemove(id)}
    icon={<CloseCircleOutlined />}
  />
</div>
<Field
  name={`components.${id}.text`}
  label={'Текст'}
  component={renderTextArea}
  parsedData={parsedData}
/>
<Field
  name={`components.${id}.posX`}
  label={'X'}
  component={renderNumberInput}
/>
<Field
  name={`components.${id}.posY`}
  label={'Y'}
  component={renderNumberInput}
/>
<Field
  name={`components.${id}.font`}
  label={'Позмір'}
  component={renderNumberInput}
/>
<Field
  name={`components.${id}.margin`}

```

```

        label={'Міжстрокова відстань'}
        component={renderNumberInput}
      />
      <Field
        name={`components.${id}.colour`}
        label={'Колір'}
        component={renderColourInput}
      />
    </div>
  )
  default:
    return "";
}

}): (
  <div className="no-data-container">
    Компоненти відсутні
  </div>
)
}
</form>
)
}

export default reduxForm({
  form: 'drawComponents', // <----- same form name
  destroyOnUnmount: false, // <----- preserve form data
  forceUnregisterOnUnmount: true, // <----- unregister fields on unmount
})(DrawComponentsList)

```

ДОДАТОК Д
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ ДЛЯ ГЕНЕРУВАННЯ СЕРТИФІКАТІВ ДЛЯ УЧАСНИКІВ МІЖНАРОДНИХ ПРОЄКТІВ

Виконав:

Студент групи КНТ-118сп

О.Г. Цибульський

Керівник:

Доцент, к.т.н

О.В. Шитікова

Рисунок Д.1 – Слайд № 1

2

Мета, об'єкт, предмет дослідження

- **Об'єкт дослідження** – процес автоматичної генерації документів.
- **Предмет дослідження** – веб-застосування для генерування сертифікатів.
- **Мета роботи** – покращення адміністративних процесів керування міжнародними проектами за рахунок генерування відповідних грамот та сертифікатів у форматі PDF документів.
- **Актуальність та проблематика роботи** полягає в тому, що більшість міжнародних проектів та організацій зустрічаються з проблемою створення та заповнення великої кількості екземплярів сертифікатів для всіх учасників програм та подій. Чим більша кількість учасників, тим більше часу виділяється для створення та заповнення сертифікатів власноруч. Це призвело до необхідності повної автоматизації заповнення сертифікатів за короткий термін часу, не зважаючи на кількість учасників.

Рисунок Д.2 – Слайд № 2

Постановка завдання до роботи

Для виконання поставленої мети потрібно виконати наступні завдання:

- створити метод для завантаження та відображення документу в ІС;
- створити метод для завантаження та перегляду даних текстового CSV файлу;
- створити метод для створення та організації ТШ для завантаженого документу;
- створити метод відображення ТКШ в області відображення завантаженого документу;
- створити метод обробки тексту КТШ, з використанням синтаксичного методу обробки;
- створити метод відображення кожного із згенерованих екземплярів документа;
- створити метод завантаження усіх згенерованих екземплярів документа у пам'яті комп'ютера.

Рисунок Д.3 – Слайд № 3

Огляд існуючих методів

Особливість	Синтаксичний метод	Метод використання структурного файлу
Простота використання	+	–
Необхідність додаткової компіляції	–	+
Швидкість обробки	+	±
Гнучкість використання	+	+
Надійність стійкості шаблону	–	+

Рисунок Д.4 – Слайд № 4

Огляд існуючих мов програмування

Особливість	JavaScript	Python	PHP
Мова створювалася спеціально для веброзробки	+	–	+
Кросплатформеність	+	+	+
Швидка інтерпретація коду	++	–	+
Простота вивчення та розробки	–	++	+
Виробничість(продуктивність роботи)	+	–	++
Наявність бібліотек та фреймворків з повною документацією	+	+	–
Наявність великої кількості методів обробки даних	+	++	+
Паралельне програмування	+	+	+
Можливість інтеграції та взаємодії з іншими мовами	++	+	+

Рисунок Д.5 – Слайд №5

Огляд існуючих середовищ розробки

Критерії порівняння	Visual Studio Code	WebStorm
Кросплатформеність	+	+
Доступність	++	±
Наявність автодоповнення	+	+
Наявність автоматичного закриття дужок, брекетів	+	+
Підтримка репозиторіїв	+	+
Наявність «plugin manager»	+	–
Простота встановлення	+	+
Підтримка аналізу коду	+	+
Підтримка рефакторингу коду	+	+
Наявність інтерактивної консолі	+	+
Зручний користувацький інтерфейс	++	±
Швидкість модернізації	++	+

Рисунок Д.6 – Слайд № 6

Схема структури ПП

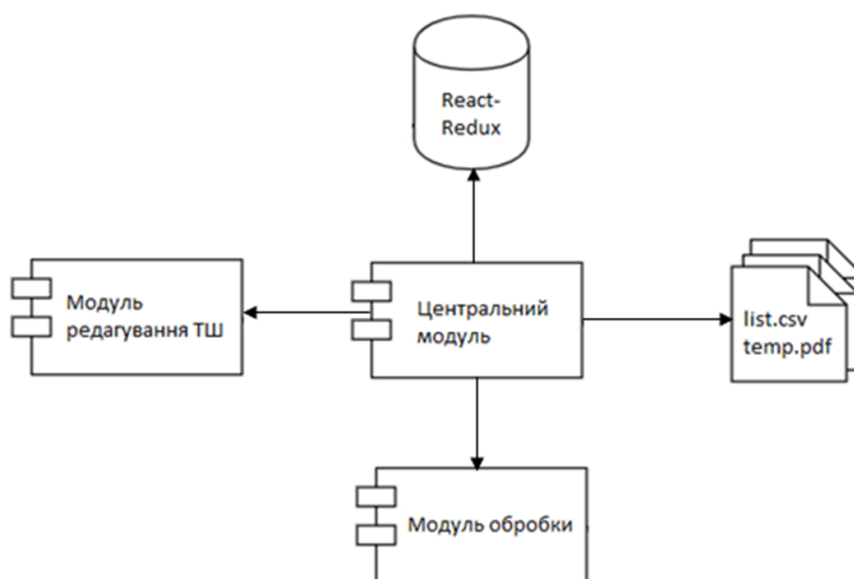


Рисунок Д.7 – Слайд № 7

Графічний інтерфейс вебзастосунку



Рисунок Д.8 – Слайд № 8

Графічний інтерфейс вебзастосунку

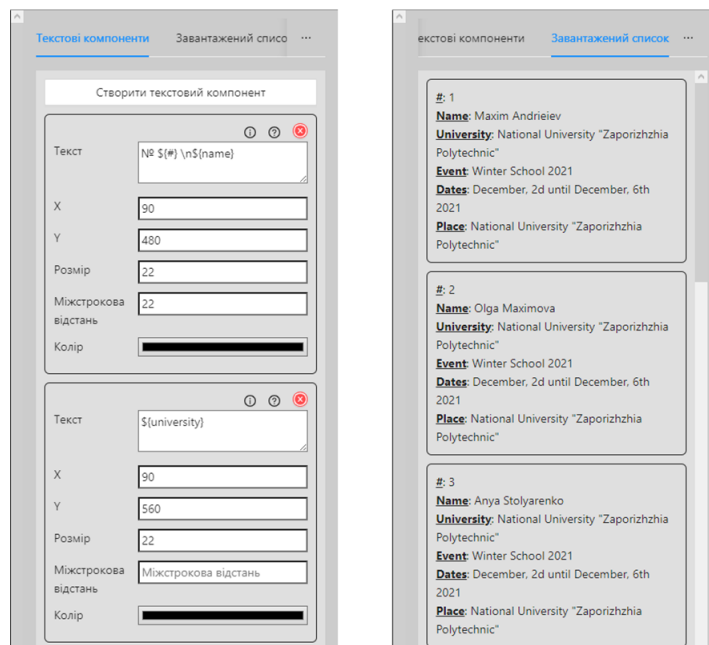


Рисунок Д.9 – Слайд № 9

Графічний інтерфейс вебзастосунку

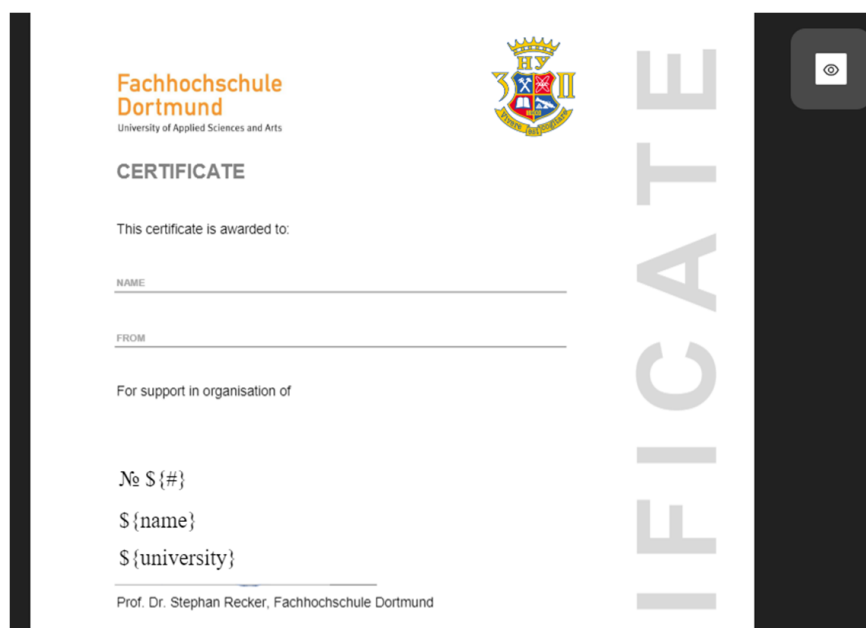


Рисунок Д.10 – Слайд № 10

Графічний інтерфейс вебзастосунку

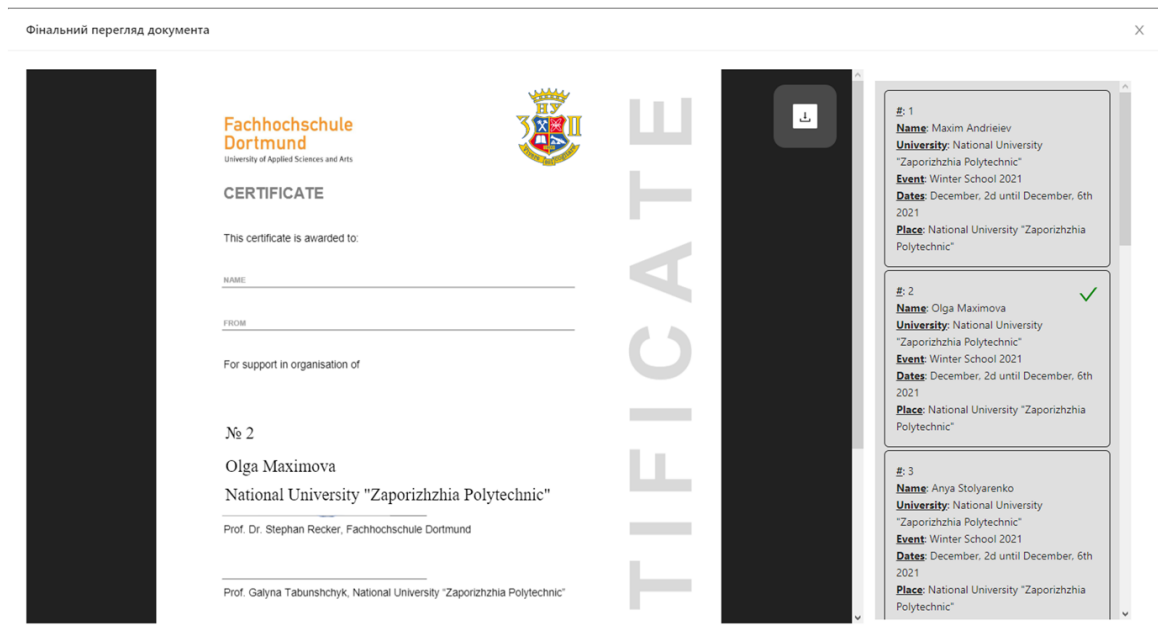


Рисунок Д.11 – Слайд № 11

Висновки

Під час виконання роботи було виконано дослідження предметної області та вивчено проблематику поставленої задачі, доведено актуальність її вирішення.

Було розроблено алгоритм обробки текстового шаблону документу з використанням синтаксичного методу аналізу тексту.

В результаті роботи було розроблено програмну систему для автоматизації процесу генерування сертифікатів міжнародних проектів шляхом розробки алгоритму створення текстового шаблону та використання синтаксичного методу для аналізу та перетворення текстового шаблону в окремий екземпляр.

Рисунок Д.12 – Слайд № 12