

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ
Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту
Завідувач кафедри
_____ Олена ОЛЬХОВСЬКА
(підпис)
«__»_____202__ р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему
**« РОЗРОБКА WEB-СЕРВІСУ ДЛЯ ГЕНЕРАЦІЇ PDF-
СЕРТИФІКАТІВ»**

зі спеціальності 122 «Комп'ютерні науки»
освітня програма «Комп'ютерні науки»
ступеня магістра

Виконавець роботи Орлова Дар'я Ігорівна

_____ «__»_____202__ р.
(підпис)

Науковий керівник к. ф.-м. н., Олексійчук Юрій Федорович

_____ «__»_____202__ р.
(підпис)

Рецензент

ПОЛТАВА 202__

АНОТАЦІЯ

Записка: 60 с., 34 рис., 1 додаток (на 93 сторінках), 34 джерел.

Мета роботи – розробка функціоналу для генерації pdf - сертифікатів з подальшою їх розсилкою електронною поштою.

Об'єкт розробки – web-сервіси для автоматизації організації освітнього процесу.

Методи дослідження та інформаційне забезпечення – аналіз, порівняння, узагальнення, моделювання, комерційне інтегроване середовище розробки для різних мов програмування від компанії JetBrains - IntelliJ IDEA та об'єктно-орієнтована мова програмування Java.

Результати дослідження, Зроблено огляд відомих сервісів генерації сертифікатів з розсилкою, виявлено позитивні та негативні сторони. Розроблено алгоритм елементів програмного забезпечення для генерації pdf - сертифікатів, побудовано блок - схему. Програмно реалізовано функціонал, в якому можна генерувати pdf - сертифікати та розсилати їх електронною поштою.

Рекомендації щодо використання результатів дослідження, програмний продукт, що реалізує web-сервіс для генерації pdf-сертифікатів впровадити в освітній процес ПУЕТ.

Наукова апробація. За матеріалами роботи опубліковано тези «Бібліотеки для генерації pdf-сертифікатів та ознайомлення з науковою публікацією схожої тематики до «Розробки web-сервісу для генерації pdf-сертифікатів»», «Проектування та розробка web-сервісу для генерування та розсилки PDF-документів», «Подібні сервіси для генерації pdf-сертифікатів», «Розширення функціоналу програмного забезпечення для генерації pdf-сертифікатів», «Дистанційна освіта в ПУЕТ», «Створення сертифікатів та унікальність кодів у web-сервісі для генерації pdf-сертифікатів».

Ключові слова: pdf-документ, java-бібліотека, itextpdf, java, backend, frontend, електронні сертифікати, генерація, web-сервіс, унікальні ідентифікатори, верифікація, Rest Api, Spring framework.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	5
ВСТУП	7
1. ПОСТАНОВКА ЗАВДАННЯ	11
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	12
2.1. Платформи навчання	12
2.2. Подібні сервіси.....	14
2.2.1. DiMaker	14
2.2.2 Canva	16
2.2.3 VistaCreate	17
2.2.4 Переваги та недоліки.....	18
2.3.Необхідність та актуальність теми.....	25
РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА	28
3.1. Стек технологій	28
3.2. Алгоритм роботи.....	33
3.3. Блок - схема	38
РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА	43
4.1. Опис програмної реалізації.....	43
4.2. Опис роботи.....	46
ВИСНОВКИ	57
РЕКОМЕНДАЦІЇ.....	60
СПИСОК ЛІТЕРАТУРИ.....	61
ДОДАТОК А. ВИХІДНИЙ КОД ПРОГРАМИ	64

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, термінів
Стартери	Набір зручних дескрипторів залежностей, які можна включити у свою програму
Стек технологій	Набір технологій, на основі яких розробляється сайт або програма.
Фреймворк	Це готова інфраструктура програмного рішення, яка забезпечує програмістам зручні умови для швидкої та ефективної розробки нових програмних продуктів. Цей програмний каркас дозволяє зосередитися на розробці конкретної функціональності, замість витрачання часу на повторювані задачі, такі як розробка базових функцій та компонентів. За допомогою фреймворка, розробники можуть значно зменшити час розробки та витрати на створення програмних продуктів, тим самим забезпечуючи більш ефективну та успішну роботу.
FinTech	Фінансові технології, які допомагають фінансовим службам та компаніям керувати фінансовими аспектами бізнесу.
JVM	Java Virtual Machine (Віртуальна машина Java)
Spring	Це потужний фреймворк на мові Java, що дозволяє ефективно

	розробляти веб – застосування
JPG	Один з найпоширеніших форматів стиснення растрових зображень. Використовується для фотографій та реалістичних зображень. Має стиснення з втратами якості.
PNG	Растровий формат з втратним та безвтратним стисненням. Часто використовується для зберігання зображень з прозорим фоном.
PDF	Поширений формат для зберігання та обміну документами з можливістю вбудовувати raster та vector графіку. Підтримує текст, зображення, гіперпосилання.
TIFF	Гнучкий растровий формат файлів зображень, що підтримує безвтратне стиснення даних. Часто застосовується в поліграфії.
PSD	Власний формат редактора Adobe Photoshop для збереження шарів та ефектів зображення. Використовується для подальшого редагування та налаштування графіки.
SMTP	Це протокол для передачі електронної пошти через Інтернет. Розшифровується як Simple Mail Transfer Protocol, що в перекладі означає "простий протокол передачі пошти".

ВСТУП

У сучасному світі, де інформація має значення, а знання стають все більш важливими, важливо мати документи, які підтверджують кваліфікацію та вміння. Сертифікати - одні з найбільш цінних документів, які дають можливість як власнику, так і іншим особам швидко оцінити рівень кваліфікації.

З іншого боку, традиційні паперові сертифікати мають певні обмеження. Їх складно верифікувати, відстежувати та зберігати. Саме тому актуальності набуває створення електронних сертифікатів з можливостями автоматичної генерації, масової розсилки та перевірки достовірності. Це дозволяє підвищити зручність, мобільність та захищеність сертифікатів.

Сертифікати навчання стають все більш популярними серед людей, які прагнуть покращити свої навички та відкрити нові можливості на ринку праці. Отримання сертифікату є найкращим способом підвищити кваліфікацію та збільшити свої шанси на просування по службі та в кар'єрі.

Водночас, традиційні паперові сертифікати мають свої недоліки: складно перевірити їх справжність, вони можуть бути втрачені або пошкоджені. Саме тому актуальним є створення електронних сертифікатів - зручних для зберігання, передачі та перевірки. Автоматизовані системи дозволяють швидко генерувати, розсилати та верифікувати такі сертифікати.

Сертифікати про навчання можуть стати джерелом мотивації для досягнення довгострокових цілей та дозволяють відстежувати власний прогрес і розвиток. Вони також дають зрозуміти, що навчання є важливим та дозволяють отримати визнання за зусилля, які вкладаються в професійний розвиток.

Водночас, традиційні паперові сертифікати не завжди забезпечують зручний доступ та можливості перевірки достовірності. Саме тому електронні сертифікати з функціями автоматичної генерації, зберігання та верифікації є

більш практичним рішенням. Вони полегшують процеси видачі, обміну та підтвердження сертифікатів між організаторами, учасниками та роботодавцями.

Сертифікат про навчання - це не тільки зручна цифрова форма визнання, але і інструмент для самовираження та підтримки навичок в актуальному стані. Він дозволяє молодому фахівцю показати свою проактивну життєву позицію, здатність до самодисципліни та самонавчання.

Усі вони підтверджують те, що навчання є важливим елементом розвитку, який необхідний в сучасному світі. Таким чином, сертифікати про навчання є необхідними для досягнення успіху в кар'єрі.

Водночас, сертифікати мають лише ту цінність, яку ми закладаємо в них на практиці. Це лише символи досягнень, але вони повинні підкріплюватися реальними знаннями та навичками. Сам по собі цифровий або паперовий документ мало значить без відповідного рівня кваліфікації.

Не менш важливими є можливості перевірки справжності сертифікатів роботодавцями. Саме тут доцільно застосовувати сучасні технології електронного документообігу з функціями швидкої верифікації безпосередньо за унікальними ідентифікаторами. Це підвищує захищеність та практичну цінність сертифікатів.

Таким чином, розробка функціоналу для отримання сертифікатів є актуальною задачею. Мій застосунок дозволить швидко та автоматично генерувати та розсилати сертифікати у великій кількості.

У даній роботі вперше розроблено веб-сервіс для автоматизації процесів генерування, розсилки та верифікації електронних сертифікатів. Сервіс дозволяє спростити підготовку сертифікатів різних форматів та призначень для освітніх закладів, громадських організацій тощо.

У роботі удосконалено алгоритм створення унікальних ідентифікаторів сертифікатів з метою підвищення захищеності від потенційних фальсифікацій.

Розроблений сервіс має істотне прикладне значення, адже його впровадження дозволить автоматизувати та оптимізувати одну з найбільш

рутинних і трудомістких задач – підготовку сертифікатів різного виду. Сервіс розроблено для реалізації в Полтавському університеті економіки і торгівлі з метою генерування сертифікатів участі в різних заходах. Його доцільно застосовувати в інших освітніх і громадських організаціях задля оптимізації документообігу.

Мета курсового проєкту – розробка програмного забезпечення для генерації pdf - сертифікатів.

Об'єкт розробки – web-сервіси для автоматизації організації освітнього процесу.

Предмет розробки – web-сервіс для генерації та розсилки pdf-сертифікатів.

Методи розробки – аналіз, порівняння, узагальнення, моделювання.

Самостійний внесок – алгоритм, блок - схема та програмне забезпечення створені повністю самостійно.

Практична новизна – створений функціонал для генерації pdf - сертифікатів з подальшою їх розсилкою. Безкоштовного сервісу з даної тематики в інтернеті не виявлено. Також це дає переваги у якості, збереженні коштів та зручності. Крім того власна система управління дає змогу налагодити процеси так, як це потрібно, а не так як передбачали розробники сторонніх інструментів. Також важливу роль в цьому питанні відіграє зменшення затрат на підтримку такої системи, оскільки придбання надійних інструментів витрачає великі кошти.

Пояснювальна записка складається з чотирьох частин. В першій – постановка завдання – сформульовано технічне завдання роботи. В другій – інформаційний огляд – переглянуто стек технологій та проаналізовано переглянуті платформи, сервіси. В третій – теоретичній – викладено алгоритм

та блок - схему. В четвертій – практичній – продемонстровано, як працює програма, та описано найважливіші моменти при створенні програми. Крім того, наведено висновки та рекомендації щодо подальшого вдосконалення програми. Також додано додаток, в якому міститься додаткова інформація, а саме код програмного забезпечення.

РОЗДІЛ 1. ПОСТАНОВКА ЗАВДАННЯ

Головне завдання кваліфікаційної роботи – розробка алгоритму, блок схеми та web-сервісу для генерації pdf - сертифікатів.

Завдання кваліфікаційної роботи:

- Розглянути матеріал наукових публікацій схожої тематики, а також подібні сервіси;
- Створити алгоритм;
- Створити блок - схему;
- Розписати процес програмного забезпечення;
- Розглянути результат програмної реалізації web-сервісу.

При розробці елементів програмного забезпечення, потрібно програмно розробити такі методи:

- Створення події, в якому буде вся інформація про подію – id, версія, назва, заголовок, час, опис, додатковий текст, дата та обраний шаблон для генерації pdf.
- Створення сертифікату, в якому буде вся інформація про сертифікат – id, версія, повне ім'я, персональна інформація, дата, адреса поштової скриньки, інформація про наявність посилання, саме посилання та подія.
- Створення розсилки.
- Створення функціоналу по додаванні користувача та зміни його ролі
- Створення функціоналу по перевірці валідності сертифікату

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Платформи навчання

Прикладом навчання з отриманням сертифікатів є Coursera[27]. Це міжнародний проект, який дозволяє отримати сертифікати за успішне проходження курсів з різних предметів. Можна зареєструватися на порталі Coursera та взяти участь у курсах від провідних університетів та компаній світу.

Завдяки курсам на Coursera, можна розвивати свої професійні та особисті навички, знайомитися з новими технологіями та тенденціями, а також підвищити свою кваліфікацію. Закінчивши курс - отримується сертифікат, який можна використовувати при подальшому просуванні по кар'єрній драбині. Зовнішній вигляд сертифікату зображено на рис. 2.2 та рис. 2.3.

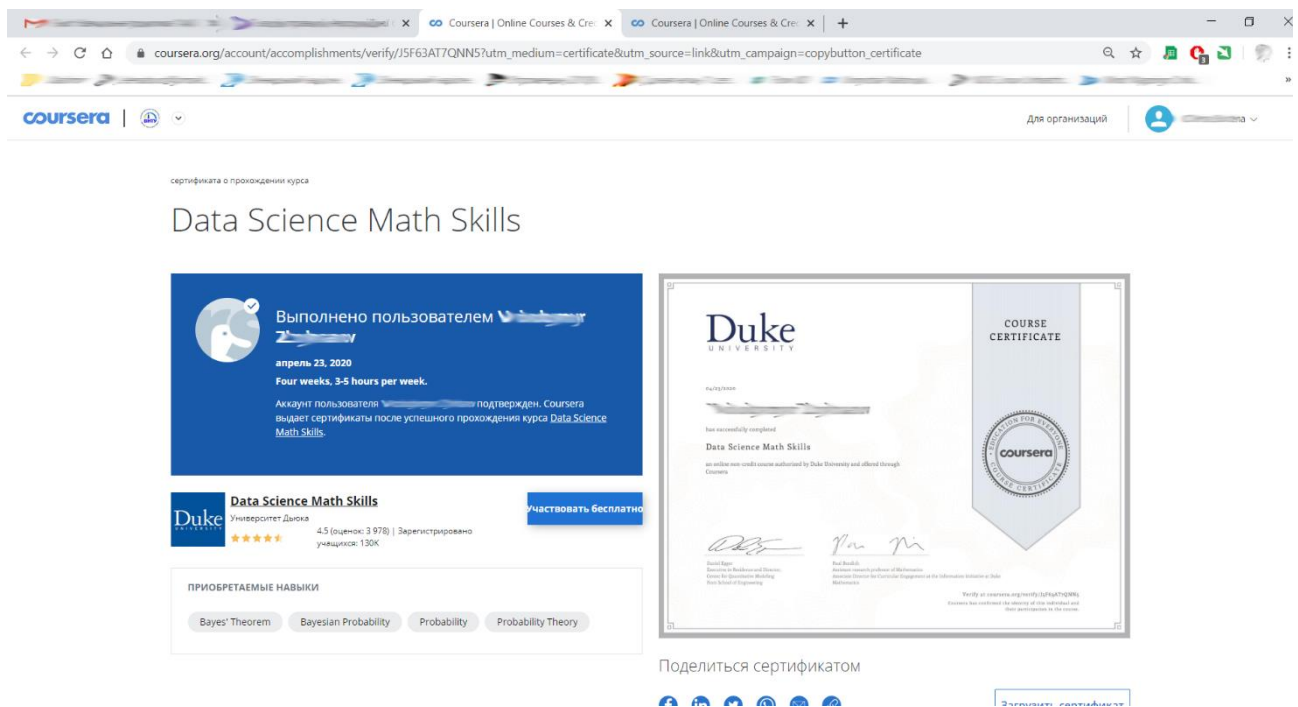


Рисунок 2.2 – Сертифікат від Coursera.

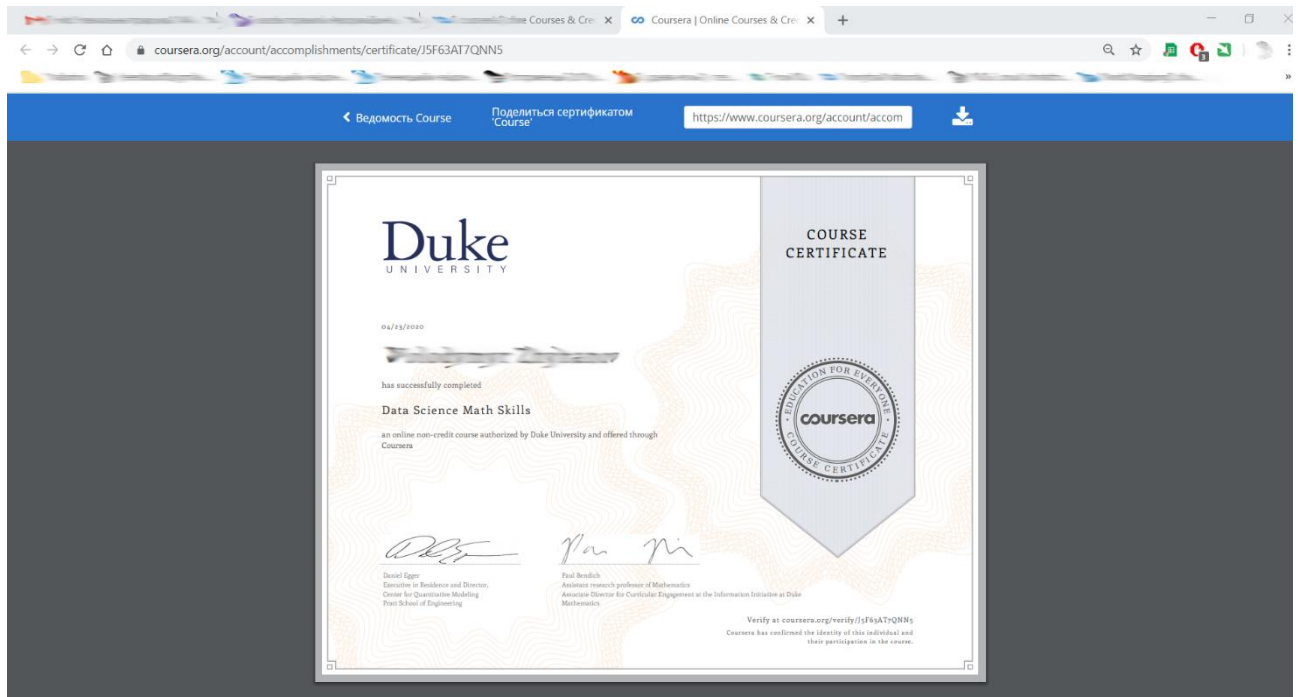


Рисунок 2.3 – Сертифікат від Coursera.

Сертифікат, отриманий після завершення курсів, зберігається у особистому кабінеті на порталі та містить інформацію про успішного завершувача, а також назву та престижний світовий університет, що підготував матеріали.

Ще однією платформою, яка допомагає отримати новітні навички, є Prometheus. Тут можна знайти більше як 250 курсів на різноманітні теми, від побудови бізнесу до програмування та місцевого самоврядування. Кожен курс має свого викладача з України чи зі світу, що гарантує актуальність та якість отримуваних знань. Крім того, Prometheus також надає сертифікати за успішне проходження курсів та лекцій з підвищення кваліфікації.

2.2. Подібні сервіси

2.2.1. DiMaker [19]

Спочатку створюється документ необхідного формату, завантажується тло, потім потрібно написати текст та розставити змінні.

Можна додавати стільки змінних, скільки потрібно і в них будуть підставленні персональні дані із таблиці.(рис. 2.4)

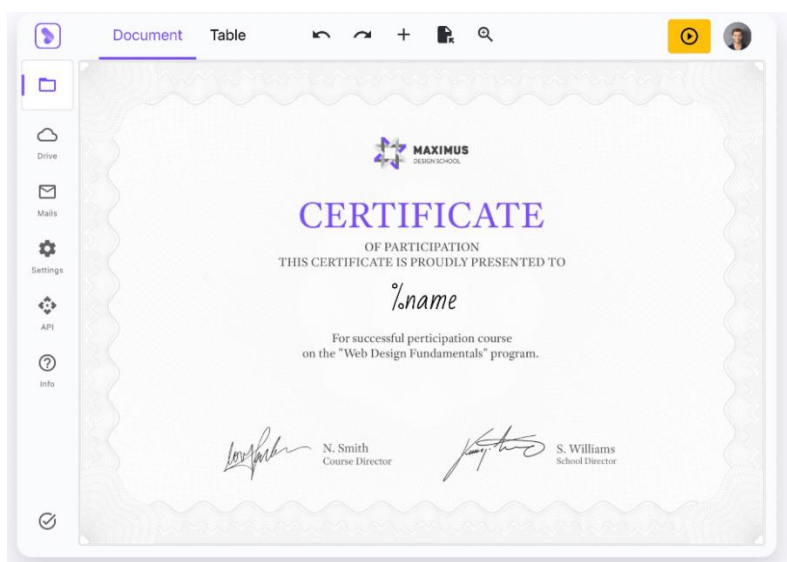


Рисунок 2.4 – Початок створення сертифікату у DiMaker.

Далі потрібно заповнити таблицю

Це робиться вручну чи можна скопіювати дані із табличного редактора.

Якщо рядків багато, можна завантажити файл формату CSV або XLS (Microsoft Excel) або завантажити з Google Таблиць.(рис. 2.5)

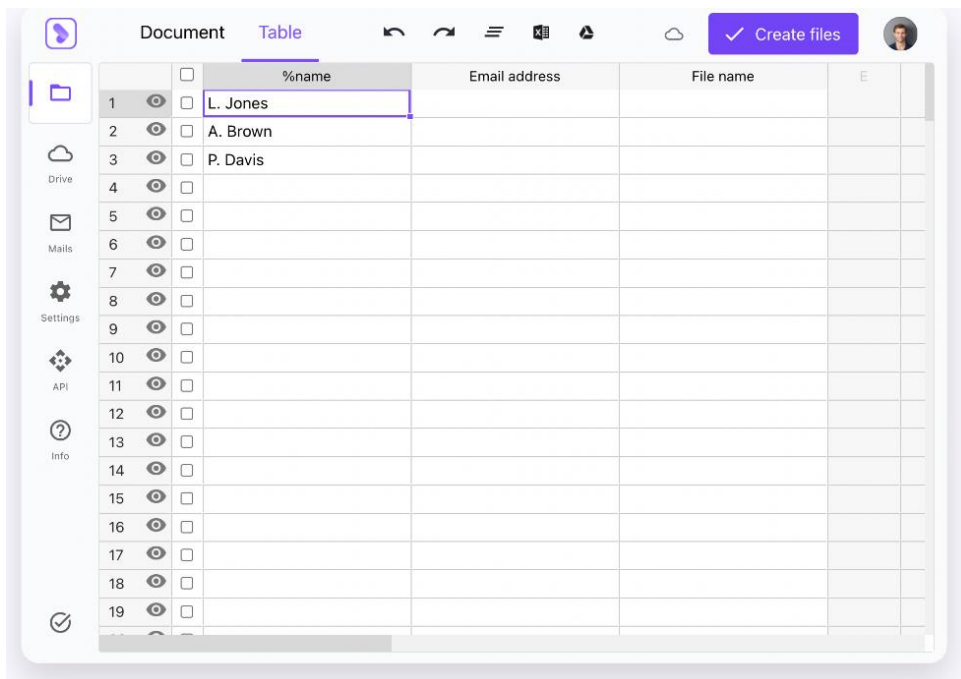


Рисунок 2.5 – Заповнення таблиці у DiMaker.

Далі можна вже отримати файли. Файли створюються у фоновому режимі. За один раз можна створити необмежену кількість файлів.

Після створення файли можна завантажити одним архівом чи багатосторінковим PDF - файлом.(рис. 2.6)

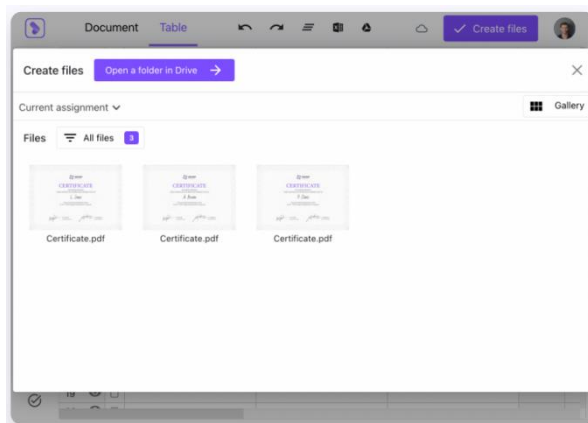


Рисунок 2.6 – Отримання файлів у DiMaker.

Далі відбувається розсилка листів. Для кожного файлу вказується адреса електронної пошти. Після розсилання відображається статус доставлення та прочитання кожного листа.

Текст листа, оформлення, відправника можна легко налаштувати. Також є можливість підключити свій домен для розсилання листів.(рис. 2.7)

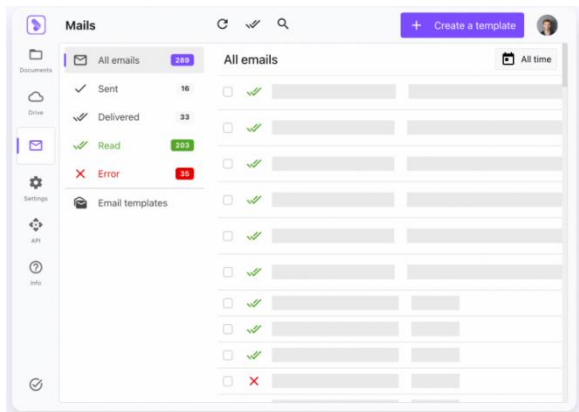


Рисунок 2.7 – Розсилка листів у DiMaker.

2.2.2. Canva [20]

Створювати можна дійсно легко та швидко. Є можливість вибрати з великого вибору готових шаблонів та легко налаштувати їх під свої потреби за допомогою інструментів перетягування. Без досвіду у дизайні можна створити привабливий та професійний сертифікат за лічені хвилини.

Як створити сертифікат

Спочатку потрібно ввести «сертифікати» в поле пошуку і потім перейти до створення дизайну.(рис. 2.8)

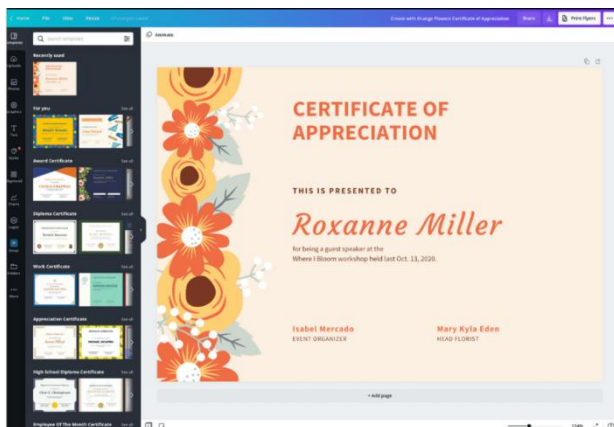


Рисунок 2.8 – Сторінка редагування в Canva.[28].

Далі треба переглянути різні стилі та теми дизайну сертифікатів та обрати ті, що потрібні. Наявний вибір від сертифікатів визнання заслуг або похвальних листів до сертифікатів про завершення курсу навчання та багатьох інших.

Далі можна змінювати текст, кольори, шрифти й фон.

Також можна додавати різні елементи дизайну.

Бібліотека зображень і графіки, яка наповнена мільйонами фотографій, значків, ілюстрацій та векторних зображень допоможе це зробити.

В кінці можна зберегти дизайн у форматі PDF, JPG або PNG[29].

2.2.3. VistaCreate [21]

Потрібно зареєструватися та увійти в обліковий запис, щоб проекти зберігалися автоматично. Далі потрібно обрати потрібний формат та шаблон з тисяч доступних варіантів, щоб швидко створити професійний дизайн. (рис. 2.9)

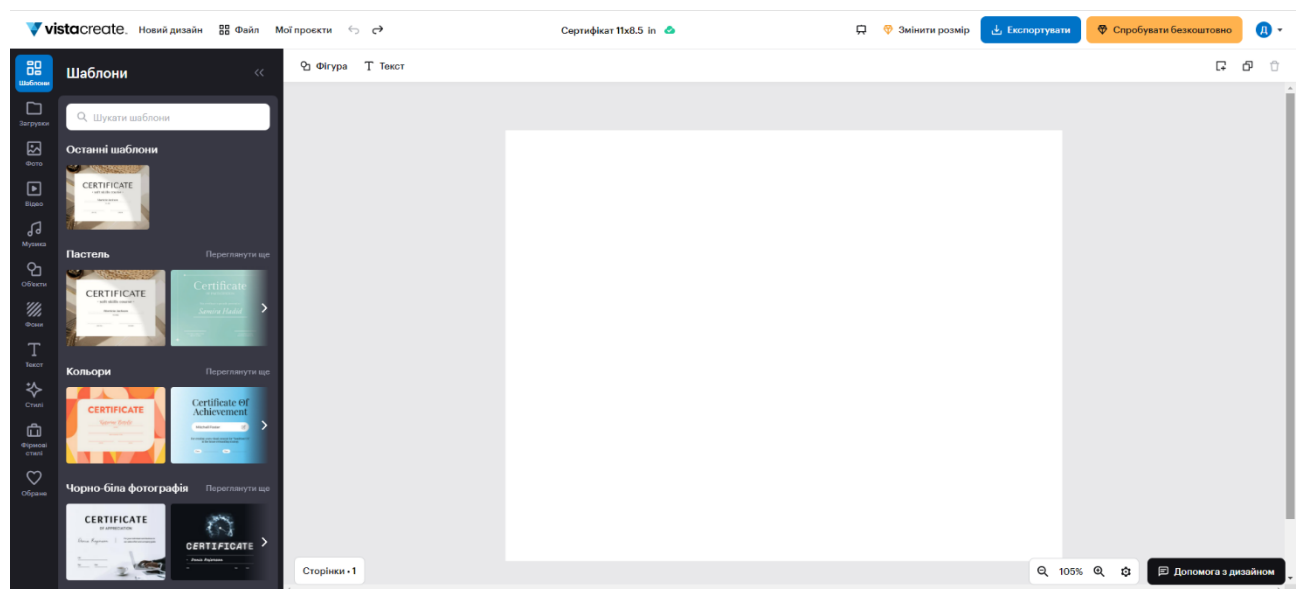


Рисунок 2.9 – Сторінка створення в VistaCreate.

Редагуючи його можна - змінювати кольори, розміри та розташування елементів. Коли дизайн готовий, його можна завантажити та роздрукувати або поділитися ним як посиланням для перегляду безпосередньо з інтерфейсу VistaCreate.

2.2.4. Переваги та недоліки

1) DiMaker

Переваги:

1)Зручний редактор:

- Можна завантажувати тло у форматі JPG, PNG, PDF, TIFF або PSD
- понад три сотні наданих шрифтів, також можна завантажувати і свої.
- будь - який розмір аркуша
- можна додавати текст, змінні, зображення, QR - код чи посилання
- Блок, в якому розміщується текст – йому можна задати розмір та розмістити в потрібному місці.

2) Простий табличний редактор

- Таблиця заповнюється даними для встановлення в змінні документа. Кожна змінна документа — окремий стовпець. На кожному рядку дані для окремого файлу.

- Кожен файл надсилається на вказану адресу електронної пошти. Якщо в осередку вказано кілька адрес через пропуск, то файл надсилається на кожну з них.

- Можна скопіювати таблицю з табличного редактора, завантажити файл CSV, XLS, XLSX або дані з Google Таблиць.

- Перед створенням файлів є можливість переглянути як буде виглядати готовий файл з даними з будь-якого рядка.

3) Створення файлів

- Якщо потрібно зберегти диплом або сертифікат, рекомендується вибрати растрований PDF - формат. У цьому випадку тло і текст будуть склеєні разом, але текст не можна буде виділити та скопіювати. Якщо потрібен подарунковий сертифікат з промо - кодом і можливістю встановити активне посилання на сайт, тоді можна вибрати опцію з окремим тлом від тексту. У цьому випадку текст можна буде виділити та скопіювати.

- друкована якість 300 DPI

- можливість встановлення розміття тексту, щоб накладений текст виглядав так само, як текст у фоновому файлі.

- 12000 файлів за годину – саме така швидкість створення файлів. За один раз можна створити необмежену кількість файлів.

- Завантажити всі файли можна одним архівом або багатосторінковим PDF після того, як вони будуть підготовлені.

4) Розсилка файлів

а) Текст листа

- Візуальний редактор

Можна красиво оформити супроводжуючий текст листа

- Вставлення посилання

Можна вставити посилання на ваш сайт, електронну пошту чи обліковий запис в соц. мережах для зворотнього зв'язку.(рис. 2.10)

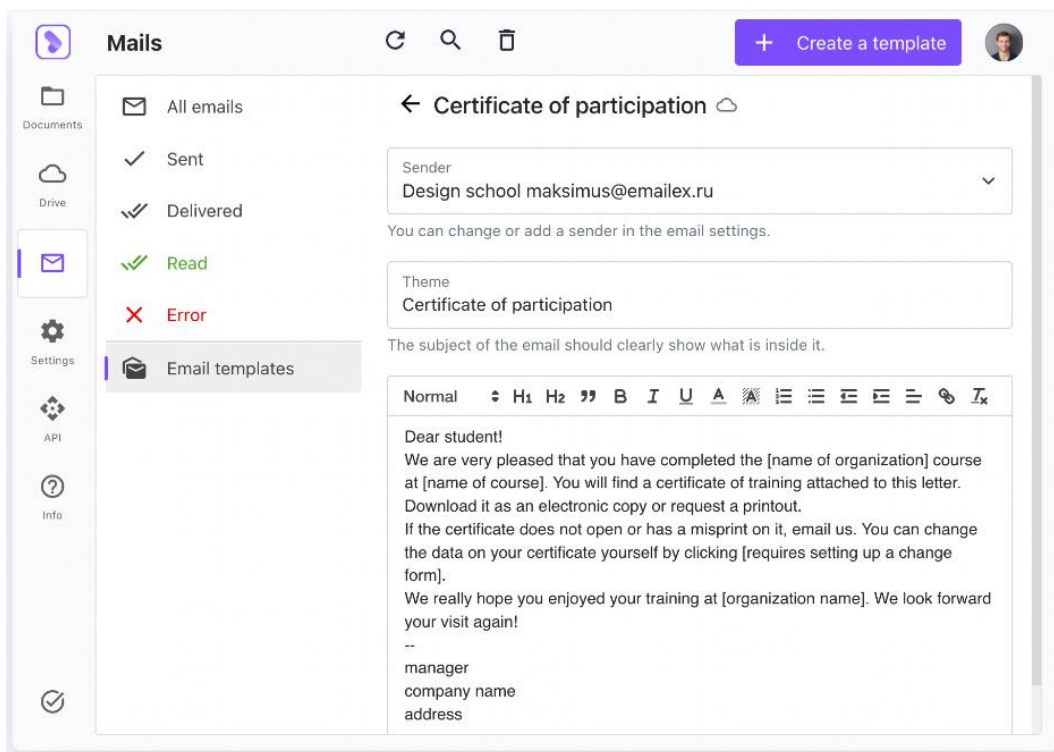


Рисунок 2.10 – Текст листа.

б) Оформлення листа

- Можна завантажити свій логотип для персоналізації оформлення (рис. 2.11)
- Якщо є шаблон листа, створений в редакторі листів, можна просто вставити HTML - код.

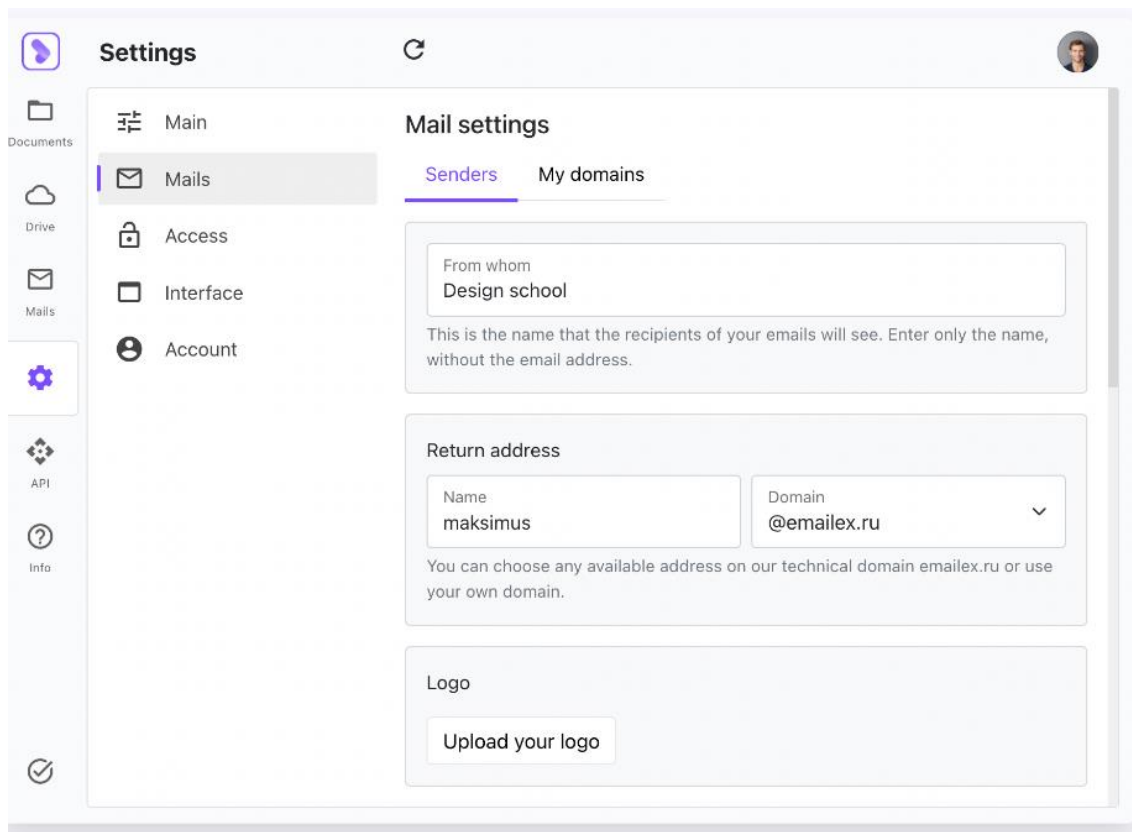


Рисунок 2.11 – Оформлення листа.

в) Відправник листа

- Можна вказати від кого лист та вказати зворотню адресу відправника.
- Можна додати свій домен, щоб зворотня адреса відправника була на вашому домені. (рис. 2.12)

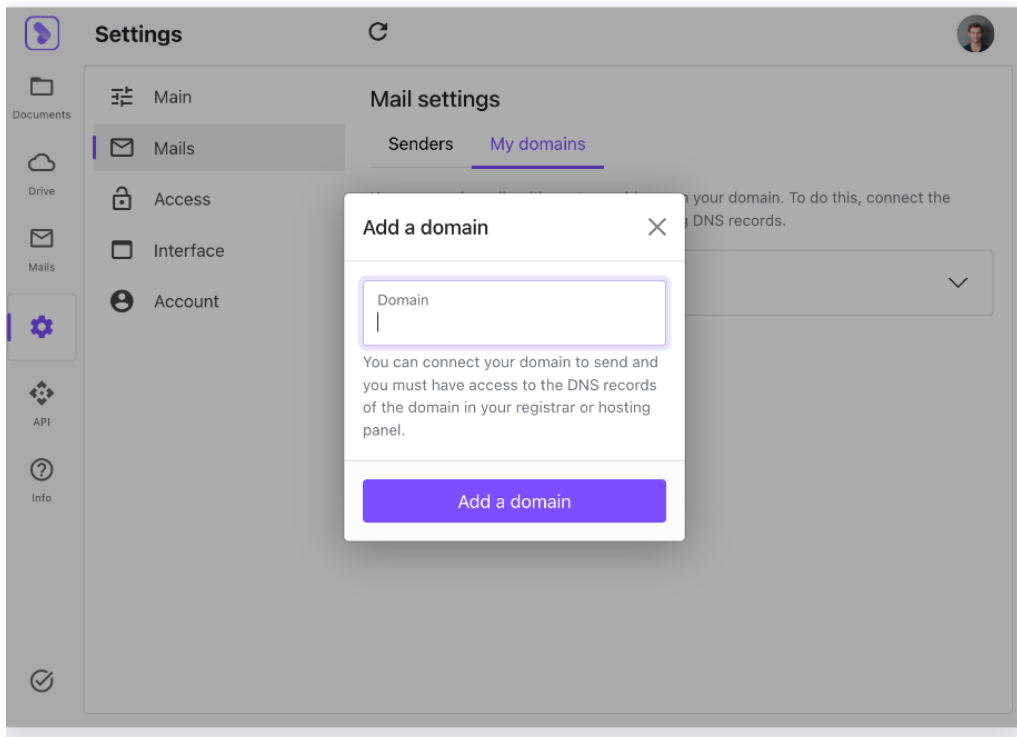


Рисунок 2.12 – Домен.

г) Час відправлення

- Листи можна надсилати відразу або через певний час
- При створенні файлів через API можна задати точний час надсилання з точністю до хвилини (рис. 2.13)

Create files [X]

Folder in Drive
Certificate

All files will be added to this folder.

Send out emails
Send emails through...

Value Interval
Minutes

Mailing after all files have been created or manually, several files can be attached to one mail.

Create (3)

Рисунок 2.13 – Час відправлення.

5) Можна додавати форми для сайту

- Віджет отримання(можна завантажувати свої дипломи чи сертифікати після введення персональних даних)
- Віджет змін (можна корегувати помилки в дипломах чи сертифікатах. Є можливість встановити ліміт на кількість та ступінь змін.)
- Віджет створення (можна автоматично створювати сертифікати, наприклад, після проходження навчання)
- Віджет опитувань (Можна збирати дані від клієнтів зі збереженням до таблиці. Ці дані можна потім використовуються для створення сертифікатів)

6) Наявна інтеграція

- з Тильдою
- з Telegram
- через API

Недоліки:

- Після тестового періоду лише 100 вітальних кредитів, які можна витратити на створення 100 файлів. Потім використання за кошти. 0.25€ за створення кожного файлу.

- Відправка електронною поштою - ще +0.25€ за кожен лист.

- До 5 Гб – безкоштовно, але якщо потрібно більше або після використання всього місця – потрібно доплачувати. При щомісячній оплаті: 25 Гб - 100.00€, 50 Гб - 200.00€, 100 Гб - 400.00€. При оплаті за рік: 25 Гб - 1000.00€, 50 Гб - 2000.00€, 100 Гб - 4000.00€.

- Баланс може «збитися в мінус», наприклад, якщо розсилається велика кількість файлів і коштів на балансі в цей час менше, ніж листів у черзі.

2) Canva

Переваги:

- а) Простота використання. Сервіс розроблений в першу чергу для новачків у дизайні та тих, для кого підготовка контенту є допоміжним напрямом діяльності, а не основним.

- б) Вибір безкоштовних функцій. Користувачам доступні понад 250 тисяч безкоштовних шаблонів графіки, кожен з яких можна змінювати: змінювати вигляд, текст, замінювати запропоновані фото на власні, додавати декоративні елементи та інше.

- в) Дизайни для будь - яких завдань. Каталог Canva має досить багатий вибір, є варіанти для пропозицій клієнтам, створення роздруківок, візитівок, виготовлення подарункових виробів, організації заходів та публікації в соціальних мережах.

- г) Можна запросити учасників до співпраці над проєктом і вже працювати разом із командою

Недоліки:

- а) Малий вибір кириличних шрифтів у Canva може стати перешкодою при створенні дизайну. Шаблони, що містять поєднання різних шрифтів,

можуть не підтримувати кириличний алфавіт, що призводить до відображення порожніх квадратів замість букв.

б) Під час використання безкоштовної версії Canva, важливо визначити розмір та формат зображення заздалегідь, оскільки зміна їх пізніше може бути неможливою. У платній версії таких обмежень немає, що дає користувачам більше можливостей для творчості.

в) Для безперебійного функціонування Canva потрібні якісні технічні параметри пристроїв та швидкий Інтернет. Рекомендується використовувати швидкий комп'ютер з достатньою оперативною пам'яттю та стабільним інтернет - з'єднанням для комфортної роботи з програмою.

г) Немає поштової розсилки

3) VistaCreate

Переваги:

- Підійде для новачків у дизайні
- Наявно понад 100 тисяч шаблонів дизайну
- Безкоштовно доступні шрифти, музика, анімація, фони та об'єкти

Недоліки:

- Змінювати розмір в безкоштовній версії не можна
- Видалити фон з шаблону та з фото в безкоштовній версії не можна
- Завантажувати у форматі HD в безкоштовній версії не можна
- Лише 10 ГБ сховища для файлів і проєктів в безкоштовній версії
- Немає поштової розсилки

2.3. Необхідність та актуальність теми

Програмування та методології, використані у розглянутих сервісах, є різними та унікальними. Тому, для досягнення оптимальних результатів, необхідно знаходити та використовувати найбільш ефективні підходи при

розробці сервісу. Це допоможе покращити процес отримання сертифікатів та досягти бажаної мети.

Під час огляду подібних сервісів не було виявлено єдиного сервісу, в якому весь представлений функціонал є безкоштовним, а ще й деякі функції, які наявні в розробленому web-сервісі, в розглянутих - взагалі відсутні.

А саме не було перевірки унікальних ідентифікаторів сертифікатів, не було можливості керувати самим процесом, а саме створювати відокремлено саму подію, додавати до неї користувачів, а саме тих, хто може керувати потім цією подією. Тобто був розроблений цілий сервіс, завдяки якому можна працювати як з цілою системою, в якій є функції, крім вище згаданих, ще й створення самих користувачів, тобто учасників події. Це все є інтегрованим по функціоналу web-сервісом для роботи з розсилкою сертифікатів, який можна підлаштувати під себе.

Було обрано pdf-формат, бо він є одним із популярних форматів для створення сертифікатів та інших документів. Використання саме pdf-документів стає кроком до створення екологічно збережливого середовища, зменшуючи використання паперу та сприяючи збереженню природних ресурсів. Крім цього, перехід до pdf дозволяє швидше, ефективніше та більш зручно обмінюватись документами. Однак процес створення та розсилки великого обсягу pdf-файлів може вимагати значних зусиль та часу. Щоб спростити цей процес, розроблено web-сервер, який автоматизує завдання щодо створення та розсилки pdf-сертифікатів. Цей сервіс надає безкоштовний та легкий у використанні функціонал, дозволяючи зосередитись на потрібних завданнях. Більш того, його гнучкість дозволяє використовувати його як самостійний інструмент або в інтеграції з іншими програмами, що робить його надзвичайно корисним та універсальним для різноманітних потреб [1-8].

Отже, проблема полягає в тому, що електронні PDF-сертифікати можна легко підробити. Це становить загрозу достовірності таких документів.

Запропонованим рішенням є використання унікальних кодів для кожного сертифікату. Ці коди дозволять перевірити справжність документа. Крім того,

потрібен надійний алгоритм генерації таких кодів. Він має забезпечити унікальність кожного коду, а також стійкість до спроб підбору чи підробки.

Впровадження системи перевірки достовірності сертифікатів на основі унікальних кодів сприятиме підвищенню надійності та захищеності електронного документообігу. Це вкрай важливо в умовах зростаючої кіберзлочинності.

Існують різні способи генерувати унікальні коди для електронних сертифікатів - від простих до більш складних.

Було застосовано посилання із унікальними кодами, а саме `site.com/VEDv132`, де `site.com` – це домен. Таким чином це ускладнює підбір чи фальсифікацію. Головне - передбачити всі сценарії використання унікальних ідентифікаторів, аби у майбутньому функціонал системи можна було легко розширювати.

Отже, з надійною та добре запрограмованою сервісною платформою можна бути впевненим в якості отриманих результатах, що підвищить шанси на успіх в обраній сфері.

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Стек технологій

Для цього проекту обрано мову програмування Java та фреймворк Spring[22]. Java (рис. 2.1) є мовою програмування та платформою обчислень, була розроблена компанією Sun Microsystems у 1995 році. Початково Java була створена як спрощена версія мови програмування C++, але відрізняється високим рівнем захисту та надійністю.

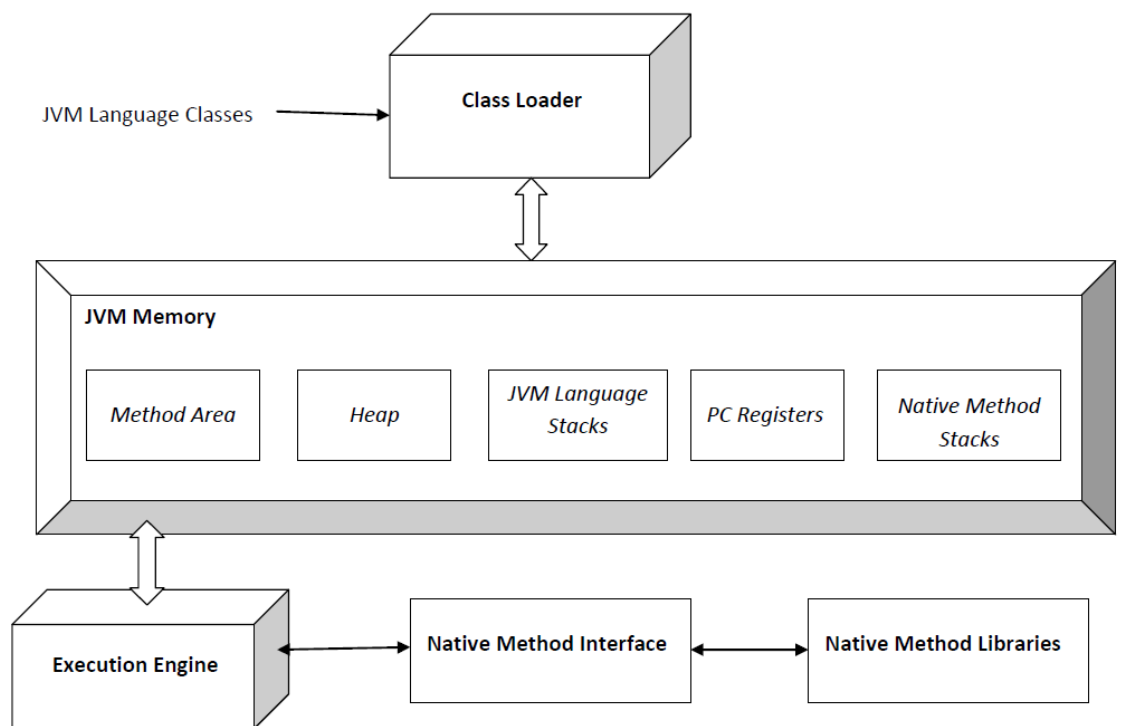


Рисунок 2.1 – Схема роботи JVM.

Для роботи потрібно встановити Java Runtime Environment (JRE), який містить наступні компоненти:

- віртуальну машину (JVM)
- компілятор
- базові бібліотеки та класи
- допоміжні бібліотеки

Віртуальна машина виконує байт - код, який був створений компілятором Java. Цей код є кросплатформеним, і тому запускається на будь - якому пристрої з встановленою JVM[26]. Отже, розроблений проект можна запускати на будь - якому пристрої або JVM.[9]

Дана мова програмування входить в список найпопулярніших мов програмування. Зазвичай на Java[24] створюють Android програми та складні відмовостійкі системи. Одним з найпопулярніших напрямків розробки на Java є створення серверних застосунків із використанням фреймворку Spring. Важливим фактором в розробці будь - якого проекту є його безпека, яка має визначальне значення для забезпечення стабільності та надійності системи. Саме тому для даного проекту було обрано мову програмування Java для розробки серверної частини. Java[25] - це мова програмування, яка в основному використовується для розробки банківських та Fintech серверних застосунків, оскільки вона забезпечує високий рівень безпеки та стійкості системи. Крім того, вибір мови програмування Java дозволить запускати серверний застосунок на практично будь - якій операційній системі, що є важливим перевагою для забезпечення гнучкості та універсальності проекту.

Фреймворк – це важлива платформа для розробки програмного забезпечення, що містить набір інструментів для створення певного типу програм, а також готові інструменти для розробки їх структури. Одна з головних відмінностей фреймворку від бібліотеки - це інверсія управління.

Працюючи у фреймворку, основний код викликає код користувача, який може виконувати конкретну поведінку, вбудовану у код фреймворку. Також фреймворк може декларувати структуру програми. Для цього проекту було обрано Spring Framework[23]. Spring Framework є фреймворком з відкритим вихідним кодом для платформи Java, який був створений Родом Джонсоном. Spring Framework є комплексною моделлю програмування та конфігурації для створення сучасних корпоративних програм на платформі Java, яка може бути використана на будь - якій платформі розгортання. Основною інфраструктурною підтримкою Spring є «підготовка» корпоративних

додатків, щоб команди могли зосередитися на бізнес - логіці на рівні програми, не вдаючись до зайвих зв'язків з конкретними середовищами розгортання. Spring також надає легкий спосіб реалізації інверсії керування, що відомий як ін'єкція залежностей (DI), де об'єкти визначають свої залежності за допомогою аргументів конструктора, заводських методів або властивостей, які встановлюються для екземпляра об'єкта після його створення. Контейнер Spring потім вводить ці залежності під час створення bean - компонента. Цей процес є зворотнім, тобто інверсією керування, де сам bean - компонент контролює створення або розташування його залежностей за допомогою прямого конструювання класів або шаблону Service Locator.

spring - boot - starter - базовий стартер, що включає підтримку автоконфігурації, логування та YAML

Spring Data JPA, частина більшої сімейства Spring Data, дозволяє легко впроваджувати репозиторії на основі JPA. Цей модуль стосується розширеної підтримки рівнів доступу до даних на основі JPA. Це полегшує створення додатків на основі Spring, які використовують технології доступу до даних.[10]

Spring Boot Starter Mail – стартер для використання Java Mail і підтримки електронної пошти Spring Framework

Spring Boot Starter Security - стартер для використання Spring Security

Spring Boot Starter Web - стартер для створення веб - додатків, включаючи RESTful, за допомогою Spring MVC. Використовує Tomcat як стандартний вбудований контейнер

Liquibase — це рішення для керування змінами схеми бази даних із відкритим вихідним кодом, яке дає змогу легко керувати версіями змін бази даних. Liquibase дозволяє будь - кому, хто бере участь у процесі випуску програми, легко:

- Усувати помилки і затримки при випуску баз даних.
- Розгортати та відкочувати зміни для певних версій, не знаючи, що вже було розгорнуто.

- Розгортати зміни бази даних і програми разом, щоб вони завжди були синхронізовані.[11]

Механізм баз даних H2

Швидка база даних SQL, яка може працювати у вбудованому або серверному режимі з підтримкою транзакцій, шифрування, повного пошуку тощо. Зберігання може бути дисковим або в пам'яті.

MySQL є вільною системою керування реляційними базами даних, розробленою компанією "ТсХ" з метою поліпшення продуктивності обробки великих обсягів даних. Як система керування базами даних з відкритим кодом, MySQL була створена як альтернатива комерційним СКБД. Початково схожа на mSQL, MySQL постійно розширювалася і зараз вона є однією з найпоширеніших систем керування базами даних. MySQL використовується переважно для створення динамічних веб - сторінок, оскільки вона має чудову підтримку різноманітних мов програмування.[12]

Проект Ломбок – дозволяє автоматично керувати ресурсами, автоматично створювати геттери, сеттери, equals, hashCode і toString тощо!

Spring Boot Starter Test - стартер для тестування програм Spring Boot з бібліотеками, включаючи JUnit Jupiter, Hamcrest і Mockito

Spring Security Test – безпека Spring

Swagger UI — це вбудоване рішення, яке значно полегшує взаємодію користувача з документацією API, створеною Swagger [13].

Swagger2 — це проект із відкритим кодом, який використовується для створення документів REST API для веб - служб RESTful. Він забезпечує інтерфейс користувача для доступу до веб - служб RESTful через веб - браузер.

Документація Swagger - це одне з найкращих рішень для документування програмного забезпечення. Вона дозволяє створювати зрозумілу та докладну документацію для програмного інтерфейсу (API) з мінімальними зусиллями. Крім того, Swagger дозволяє проводити легке тестування програми, що значно полегшує процес розробки та тестування. За допомогою Swagger, розробники можуть швидко та легко створювати та

оновлювати документацію API, забезпечуючи зрозумілі та детальні інструкції щодо використання API для користувачів. В результаті, використання Swagger дозволяє підвищити ефективність та точність процесу розробки програмного забезпечення.

Завантажувач SpringFox - документація JSON API для програм на основі Spring

Система збірки Maven дозволяє розробникам швидко створювати інтерактивні додатки на базі Spring Framework. Завдяки цьому плагіну можна легко сконфігурувати та запустити проект, а також автоматично вирішувати залежності на бібліотеки інших розробників. Maven збирає усі бібліотеки, описані у файлах POM, в один файл .jar, який містить усі залежності проекту. Завдяки цьому розробникам легше управляти проектом та дотримуватися кращих практик розробки програмного забезпечення. Також це дозволяє збільшити швидкість розробки, забезпечити кращу структуру проекту та зробити розробку проекту більш ефективною та приємною.

Для роботи з PDF найкраще підійде java - бібліотека iText. iText – це бібліотека Java, за допомогою якої можна розробляти програми, які створюють, конвертують і обробляють документи PDF [14-15].

SMTP - це протокол, який використовується для надсилання електронних листів між серверами. Він забезпечує швидку та надійну доставку повідомлень. Ось кілька ключових моментів щодо SMTP та SMTP-серверів:

- SMTP-сервер спеціалізується на передачі пошти за допомогою SMTP протоколу. Кожен сервер має унікальну адресу для ідентифікації.
- Стандартний порт для з'єднання з SMTP сервером - 25. Також часто використовуються порти 465 та 587 для захищених з'єднань.
- Існують безкоштовні та платні SMTP сервери. Безкоштовні мають обмеження на обсяги та швидкість розсилок. Платні дають більше гнучкості.
- Поштові сервіси, хостинг-провайдери та спеціалізовані transactional-сервіси надають можливість відправляти листи через SMTP.

- Правильні налаштування та використання SMTP допомагає уникнути проблем зі спамом та блокуванням IP-адреси.

Ось порівняння декількох популярних безкоштовних SMTP-сервісів для надсилання email-розсилок:

SendPulse - 12 000 листів на місяць, зручний особистий кабінет з аналітикою та готовими шаблонами.

Sendinblue - 300 листів на день, детальна статистика в реальному часі, можливість використання виділених IP.

Mailjet - 6000 листів на місяць, зручний drag-and-drop редактор для створення розсилок.

SendGrid - 40000 листів протягом 30 днів, потім 100/день, API та SMTP інтеграція.

Amazon SES - 62 000 безкоштовних листів, можлива інтеграція з AWS хмарою.

SparkPost - 15000 на місяць, аналітика в реальному часі, підтримка виділених IP (платно).

Mailgun - 10000 листів на місяць, REST API та SMTP можливості, розширена аналітика.

Основна перевага використання Gmail SMTP - це простота налаштування без додаткової реєстрації, і тому в якості SMTP було обрано Gmail.

3.2. Алгоритм роботи.

Користувач може здійснити такі основні дії:

- створити захід
- додати користувача до заходу
- отримати список заходів
- створити користувача
- змінити роль користувача
- отримати список користувачів

- додати інформацію про учасника заходу
- отримати інформацію про сертифікати
- імпортувати сертифікати з csv-файлу
- згенерувати сертифікати
- розсилати сертифікати
- завантажити сертифікати
- перевірити валідність сертифікату

Після переходу на [http://localhost:8080/swagger - ui/](http://localhost:8080/swagger-ui/) на сторінці відображаються такі методи:

a) certificate - controller

- POST

/api/v1/certificate - management/certificates

(createCertificate)

- GET

/api/v1/certificate-management/certificates/code/{code}

(getCertByCode)

- POST

/api/v1/certificate-management/certificates/csv

(importFromCSV)

- GET

/api/v1/certificate-management/certificates/date/{date}

(getAllByEventDate)

- GET

/api/v1/certificate-management/certificates/email/{email}

(getAllByCertificateEmail)

- GET

/api/v1/certificate - management/certificates/{eventId}

(getAllByEventId)

- GET

/api/v1/certificate-management/certificates/eventName/{eventName}

getAllByEventName

- GET

/api/v1/certificate-management/certificates/fullName/{fullName}

(getAllByCertificateFullName)

б) email - controller

- POST

/api/v1/email - management/send - by - certificate - id

(setEmailByCertificateId)

- POST

/api/v1/email - management/send - by - event - id

(setEmailByEventId)

- POST

/api/v1/email - management/send - simple - test

(sendTestSimpleMail)

в) event - controller

- GET

/api/v1/event - management/events

(getAll)

- POST

/api/v1/event - management/events

(createEvent)

- POST

/api/v1/event-management/events/{id}/addUser/{email}

(addUser)

- GET

/api/v1/event-management/events/date/{date}

(getByEventDate)

- GET

/api/v1/event-management/events/eventName/{eventName}

(getByEventName)

- GET

/api/v1/event - management/events/{id}

(getById)

г) pdf - controller

- GET

/api/v1/pdf - management/pdf/{certId}

(getPdf)

д) user-controller

- POST

/api/v1/user-management/change/role/{id}/{role}

(changeRole)

- POST

/api/v1/user-management/user/{email}

(createUser)

- GET

/api/v1/user-management/users

(getAllUsers)

- GET

/api/v1/user-management/users/{id}

(getUserById)

е) validator-controller

- GET

/p/{code}

(show)

Крок 1. Користувач у event - controller обирає метод POST /api/v1/event - management/events (createEvent). Натискає Try it out. Потрібно заповнити JSON. Потім потрібно натиснути Execute. При успішному виконанні буде відповідь 200 OK. В іншому випадку – 500 Error.

Крок 2. За потреби можна додати користувача. Користувач обирає метод POST addUser у event-controller. Заповнює поля. Потім потрібно натиснути

Execute. При успішному виконанні буде відповідь 200 OK. В іншому випадку – 500 Error.

Крок 3. За потреби можна створити користувача або змінити його роль. Для цього користувач виконує запит `createUser/changeRole` в `user-controller`. При успішному виконанні буде відповідь 200 OK. В іншому випадку – 500 Error.

Крок 4. Користувач у `certificate - controller` обирає метод `POST /api/v1/certificate - management/certificates (createCertificate)`. Натискає `Try it out`. Потрібно заповнити JSON. Потім потрібно натиснути `Execute`. При успішному виконанні буде відповідь 200 OK. В іншому випадку – 500 Error. Або можна виконати запит `POST importFromCSV` у `certificate-controller`.

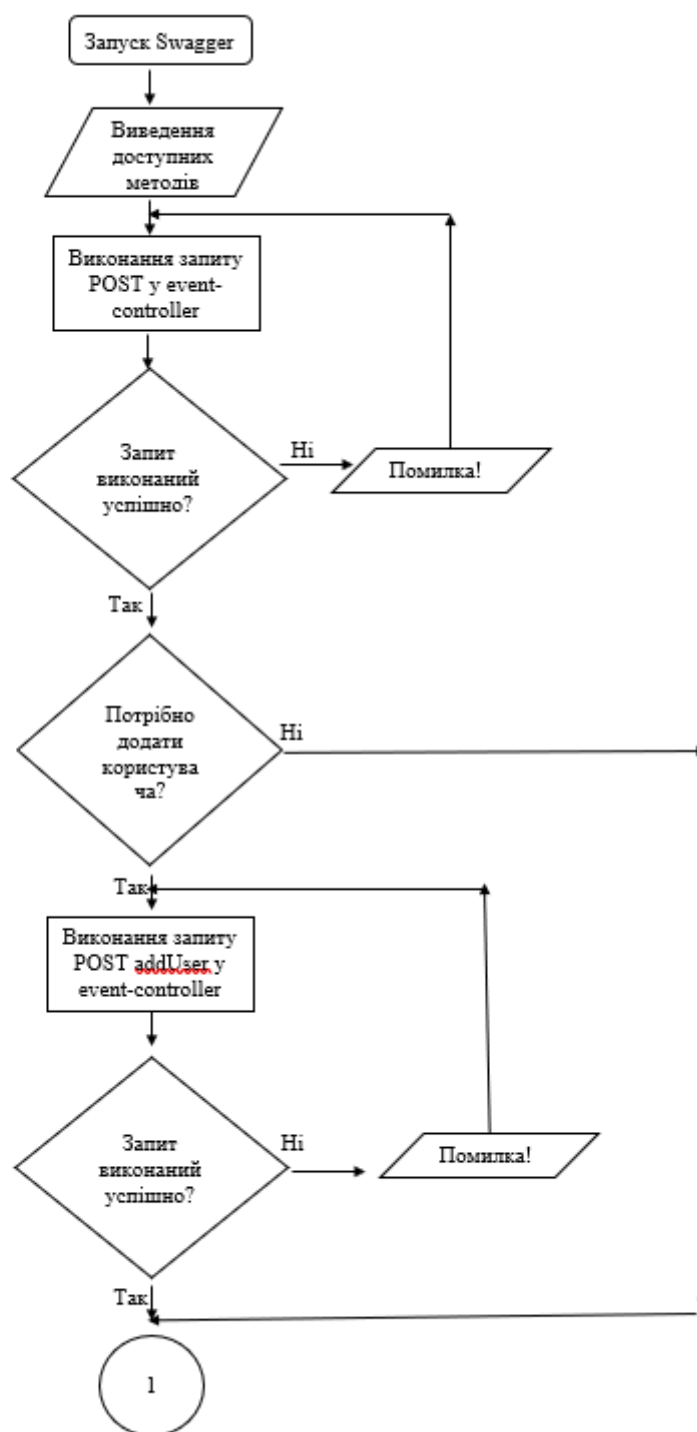
Крок 5. За потреби якщо потрібен pdf-сертифікат, то користувач виконує запит `GET getPdf` у `pdf-controller`. При успішному виконанні буде відповідь 200 OK. В іншому випадку – 500 Error.

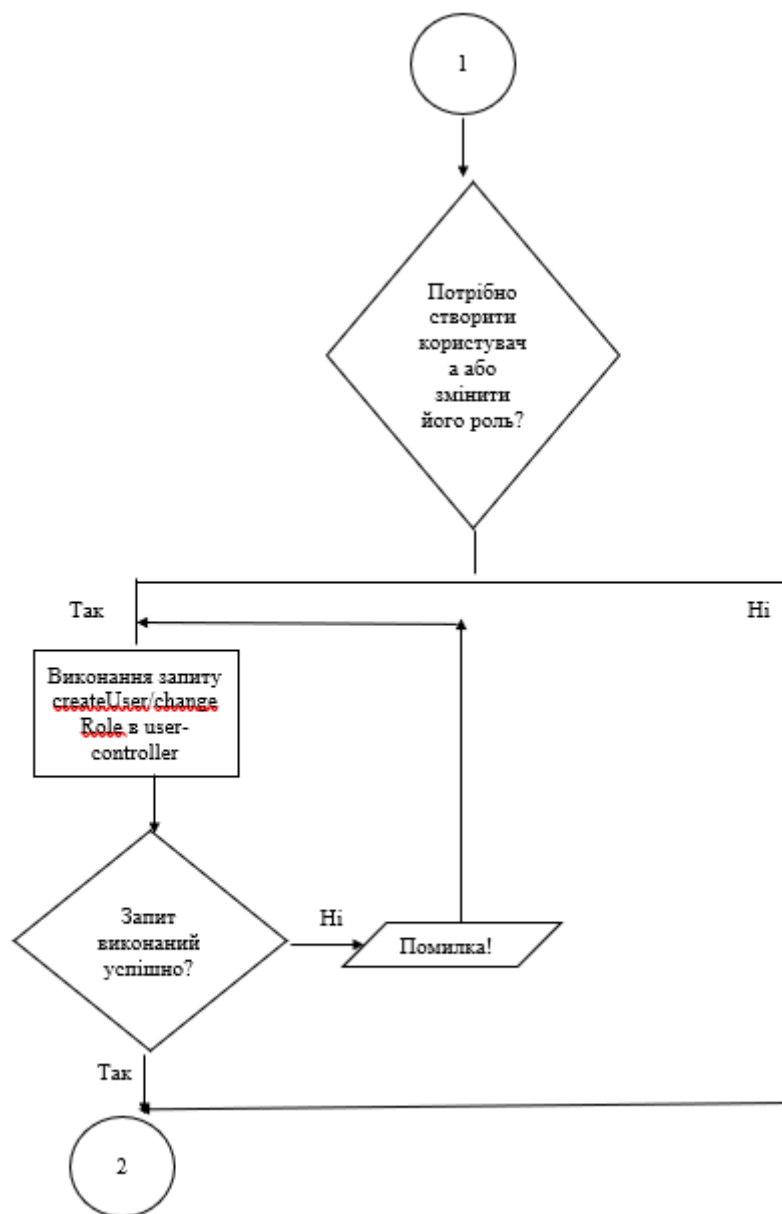
Крок 6. Користувач у `email - controller` обирає метод `POST /api/v1/email - management/send - by - event - id (setEmailByEventId)`. Натискає `Try it out`. Вводить `id`. Потім потрібно натиснути `Execute`. При успішному виконанні буде відповідь 200 OK. В іншому випадку – 500 Error.

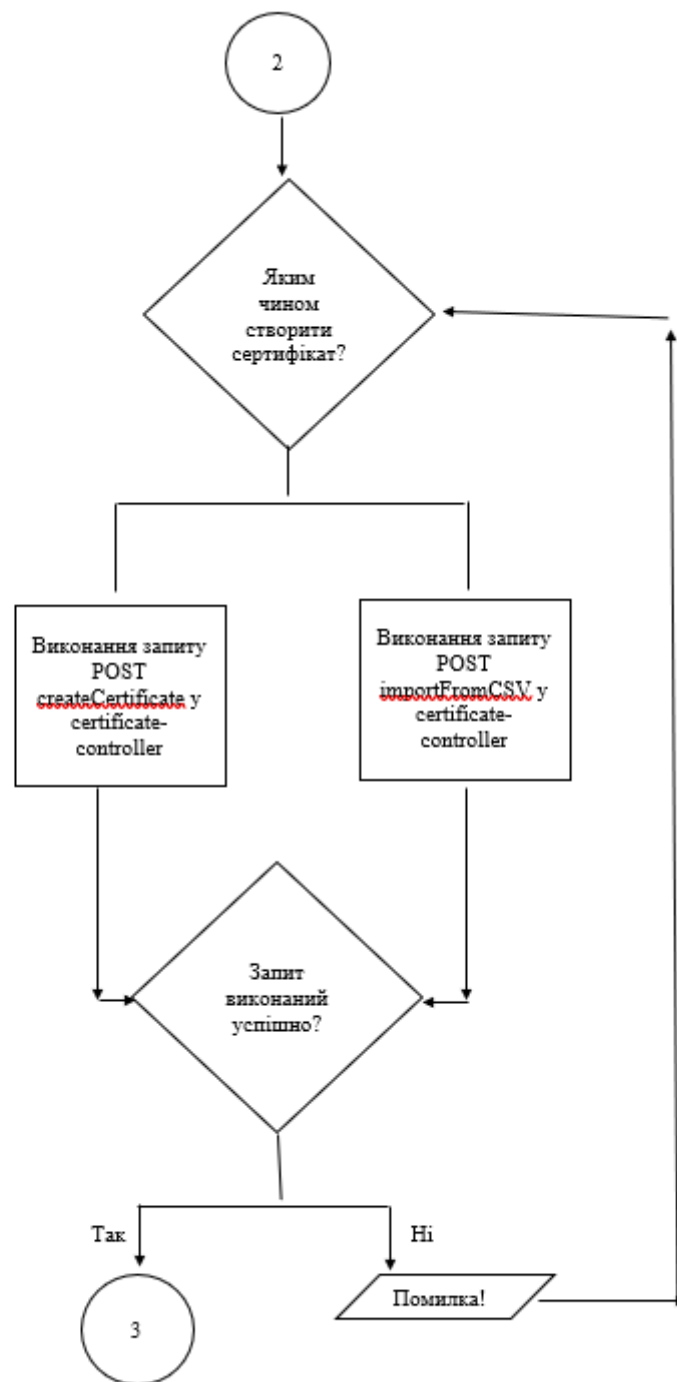
Крок 7. За потреби якщо потрібно перевірити валідність сертифікату, то користувач виконує запит `GET show` у `validator-controller`. При успішному виконанні буде відповідь 200 OK. В іншому випадку – 500 Error.

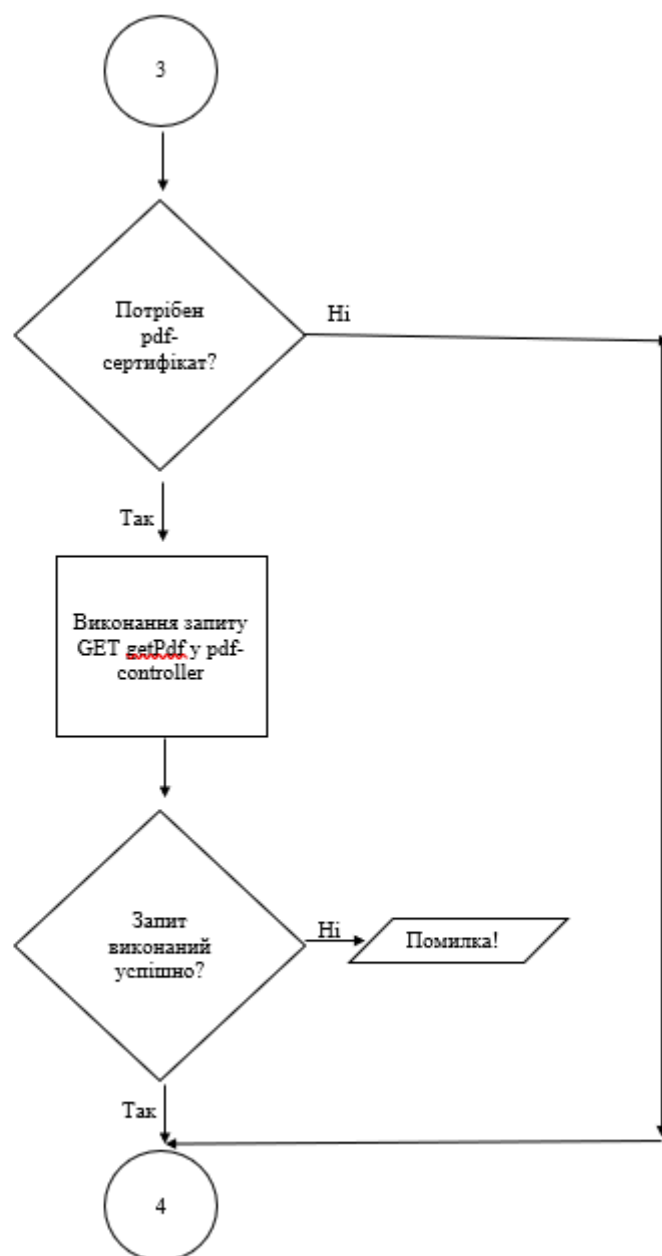
3.3. Блок - схема

Блок - схема (рис. 3.1)









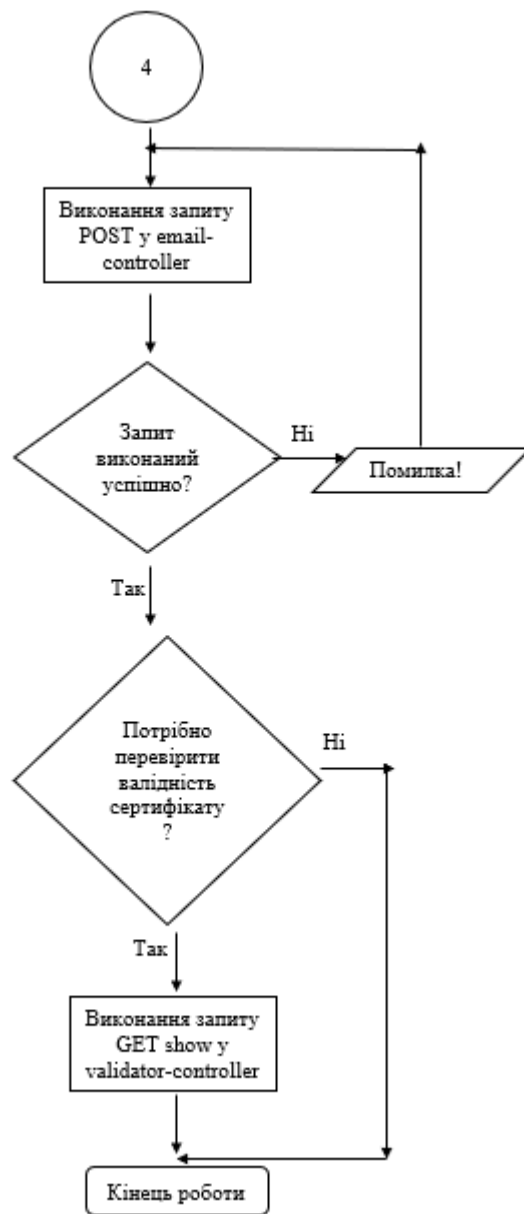


Рисунок 3.1 – Блок–схема типового алгоритму роботи із web-сервісом.

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис програмної реалізації

Web - сервіс програмно реалізовано за допомогою мови програмування Java, в середовищі IntelliJ IDEA. Створення нового проєкту відбувається на сайті <https://start.spring.io/> (рис 4.1).

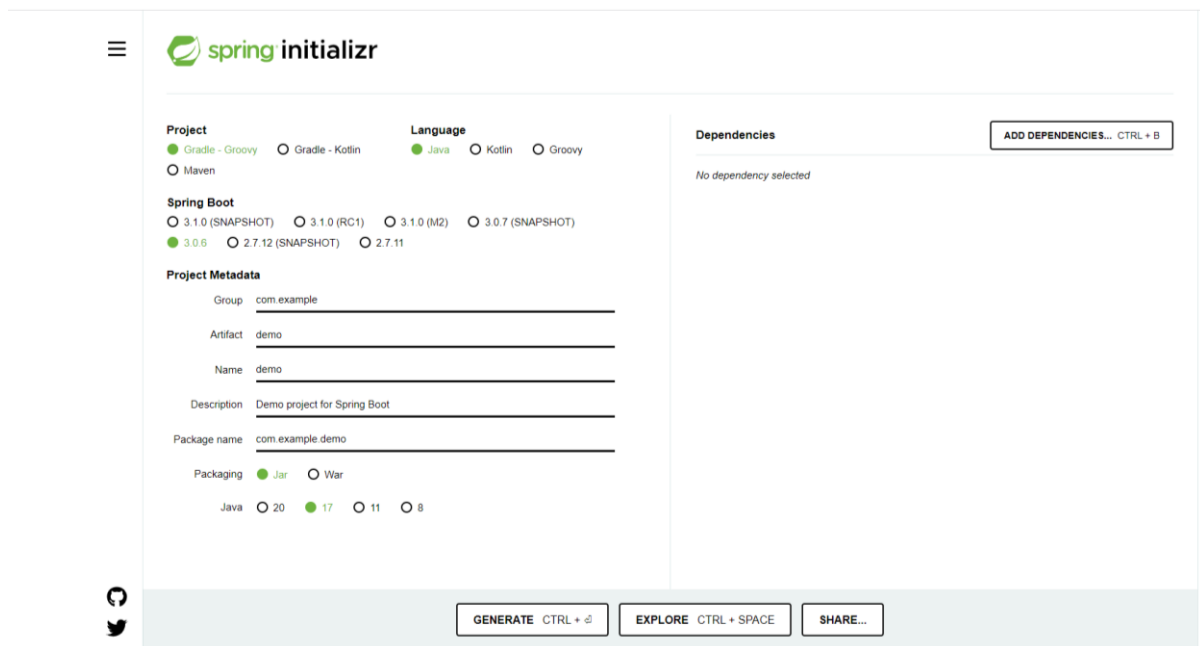
The screenshot shows the Spring Initializr web application. At the top, there is a hamburger menu icon, the 'spring initializr' logo, and a settings icon. The main content area is divided into three sections. The 'Project' section has radio buttons for 'Gradle - Groovy' (selected), 'Gradle - Kotlin', and 'Maven'. The 'Language' section has radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'. The 'Spring Boot' section has radio buttons for '3.1.0 (SNAPSHOT)', '3.1.0 (RC1)', '3.1.0 (M2)', '3.0.7 (SNAPSHOT)', '3.0.6' (selected), '2.7.12 (SNAPSHOT)', and '2.7.11'. The 'Project Metadata' section contains text input fields for 'Group' (com.example), 'Artifact' (demo), 'Name' (demo), 'Description' (Demo project for Spring Boot), and 'Package name' (com.example.demo). There are also radio buttons for 'Packaging' (.jar selected, .war) and 'Java' version (20, 17 selected, 11, 8). The 'Dependencies' section on the right has a button 'ADD DEPENDENCIES... CTRL + B' and the text 'No dependency selected'. At the bottom, there are three buttons: 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'. Social media icons for GitHub and Twitter are visible in the bottom left corner.

Рисунок 4.1 – створення проєкту.

Проєкт, що реалізовується, використовує відомий патерн проєктування Controller - Service – Repository [16-17] та складається із відповідних рівнів. На рівні entity описані сутності з бази даних у вигляді об'єктів, що забезпечує зручний доступ до даних. На рівні репозиторію описуються запити до бази даних з використанням інтерфейсу JpaRepository, що дозволяє забезпечити потрібну функціональність. Логіка роботи web - сервісу реалізована на рівні сервісів. Кожен клас - сервіс відповідає за виконання певної задачі, або кілька класів - сервісів, якщо задача велика та її доцільно розбити на підзадачі. REST - контролери відповідають за зовнішній інтерфейс та забезпечують зручний доступ до сервісів. Така архітектура проєкту дозволяє забезпечити якісну та ефективну роботу, а також зробити його більш зручним та зрозумілим для

користувачів. Тенденція останнього десятиліття в розробці програмних продуктів з API полягає у поступовому переході від технологій SOAP (Simple Object Access Protocol) та RPC (Remote Procedure Call) до технології REST (Representational State Transfer) [30-32].

REST — підхід до архітектури мережеских протоколів, які надають доступ до інформаційних ресурсів, що був описаний Роєм Філдінгом в дисертації в 2000 році [9]. Реалізація Web-сервісів REST відповідає чотирьом базовим принципам проектування:

1. явне використання HTTP-методів;
2. незбереження стану;
3. надання URI, аналогічних структурі каталогів;
4. передача даних в XML, JavaScript Object Notation (JSON) або в обох форматах [33-34].

У проєкті є такі пакети, класи, інтерфейси й перерахування:

1) api

- CertificateController.java
- EmailController.java
- EventController.java
- PdfController.java
- UserController.java
- ValidatorController.java

2) dto

- CertificateDTOInput.java
- CertificateDTOOutput.java
- EventDTOInput.java
- EventDTOOutput.java
- EventDTOOutputWithTemplate.java
- UserDTO.java

3) entity

- Certificate.java
- Event.java
- User.java

4) enums

- Template.java - перерахування
- UserPermission.java – дозволи користувачів
- UserRole.java – ролі користувачів
- UserStatus.java – статуси користувачів

5) exceptions

- CalendarException.java
- DataBaseException.java
- EmailException.java

6) mapper

- CertificateMapper.java
- EventMapper.java
- UserMapper.java

7) repo

- CertificateRepository.java - інтерфейс
- EventRepository.java - інтерфейс
- UserRepository.java - інтерфейс

8) service

- CertificateService.java
- EmailService.java
- EventService.java
- PdfService.java
- UserDetailsServiceImpl.java
- UserService.java

9) utils

- data (підпакет) – EmailAttachment.java
- CalendarUtil.java

- EmailSender.java
- EmailSenderImpl.java
- PdfCreator.java
- PdfGenerator.java

10) - PdfServiceApplication.java

11) config

- SecurityConfiguration.java

Також є файли application.properties з налаштуваннями, pom.xml з залежностями, .gitignore та .env.

Код проєкту наведений в додатку А.

4.2. Опис роботи

Після запуску Swagger користувачу відображаються методи для взаємодії (рис 4.2).

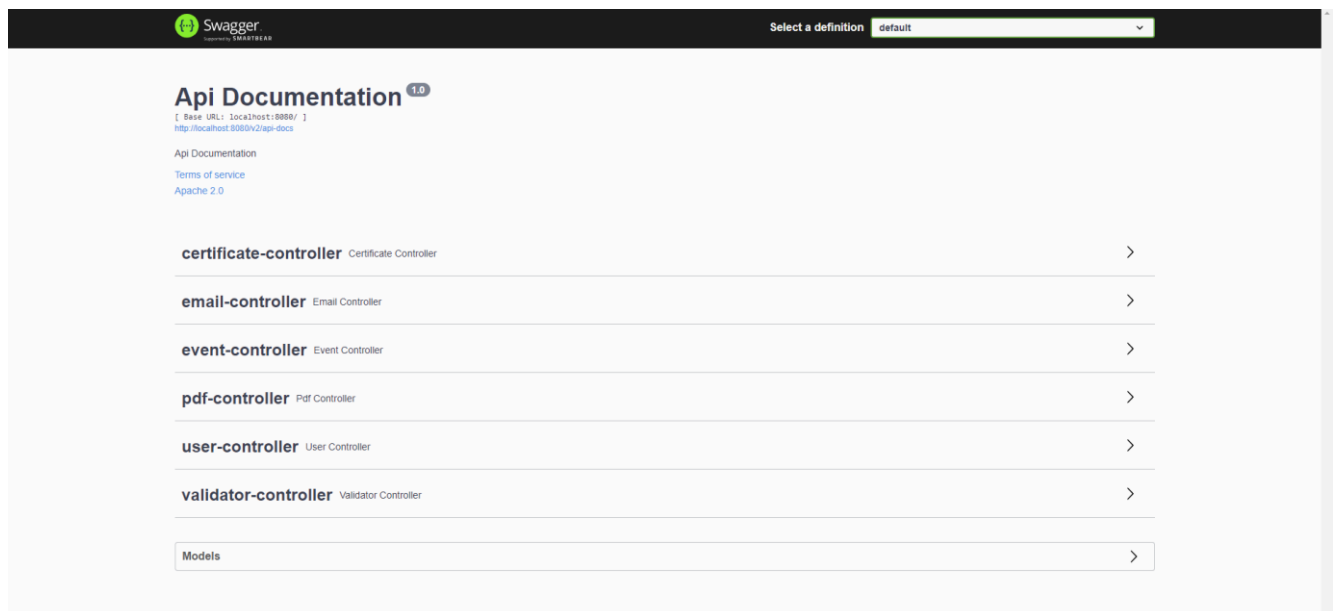


Рисунок 4.2 – стартова сторінка.

Спочатку створюється подія. Для цього користувач обирає метод POST /api/v1/event - management/events (createEvent) і натискає Try it out. Заповнює JSON (рис 4.3).

event-controller Event Controller

GET /api/v1/event-management/events getAll

POST /api/v1/event-management/events createEvent

Parameters

Name Description

eventDTO * required eventDTO

object Edit Value | Model

(body)

```
{
  "additionalText": "Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae",
  "date": "28.03.2023",
  "eventName": "Задеснеження копейнички подорож",
  "hours": 2,
  "id": 0,
  "mainText": "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.",
  "template": "TEST",
  "title": "Здегірамм гануек"
}
```

Cancel

Parameter content type application/json

Execute

Responses

Response content type */*

Рисунок 4.3 – event - controller.

Потім натискає Execute. При успішному виконанні отримує відповідь 200 OK (рис. 4.4).

Execute Clear

Responses

Response content type */*

Curl

```
curl -X POST "http://localhost:8080/api/v1/event-management/events" -H "accept: */*" -H "Content-Type: application/json" -d '{"additionalText": "Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae", "date": "28.03.2023", "eventName": "Задеснеження копейнички подорож", "hours": 2, "id": 0, "mainText": "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.", "template": "TEST", "title": "Здегірамм гануек"}'
```

Request URL

http://localhost:8080/api/v1/event-management/events

Server response

Code Details

200

Response body

```
{
  "id": 1,
  "eventName": "Задеснеження копейнички подорож",
  "title": "Здегірамм гануек",
  "hours": 2,
  "mainText": "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.",
  "additionalText": "Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae",
  "date": "28.03.2023",
  "template": "TEST"
}
```

Response headers

```
cache-control: no-cache,storemax-age=0,must-revalidate
connection: keep-alive
content-type: application/json
date: Mon, 24 Apr 2023 12:21:16 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
x-content-type-options: nosniff
x-frame-options: DENY
x-ssr-protection: 1; mode=block
```

Responses

Code Description

Рисунок 4.4 – успішне виконання.

Якщо сталася помилка – 500 Error (рис 4.5).

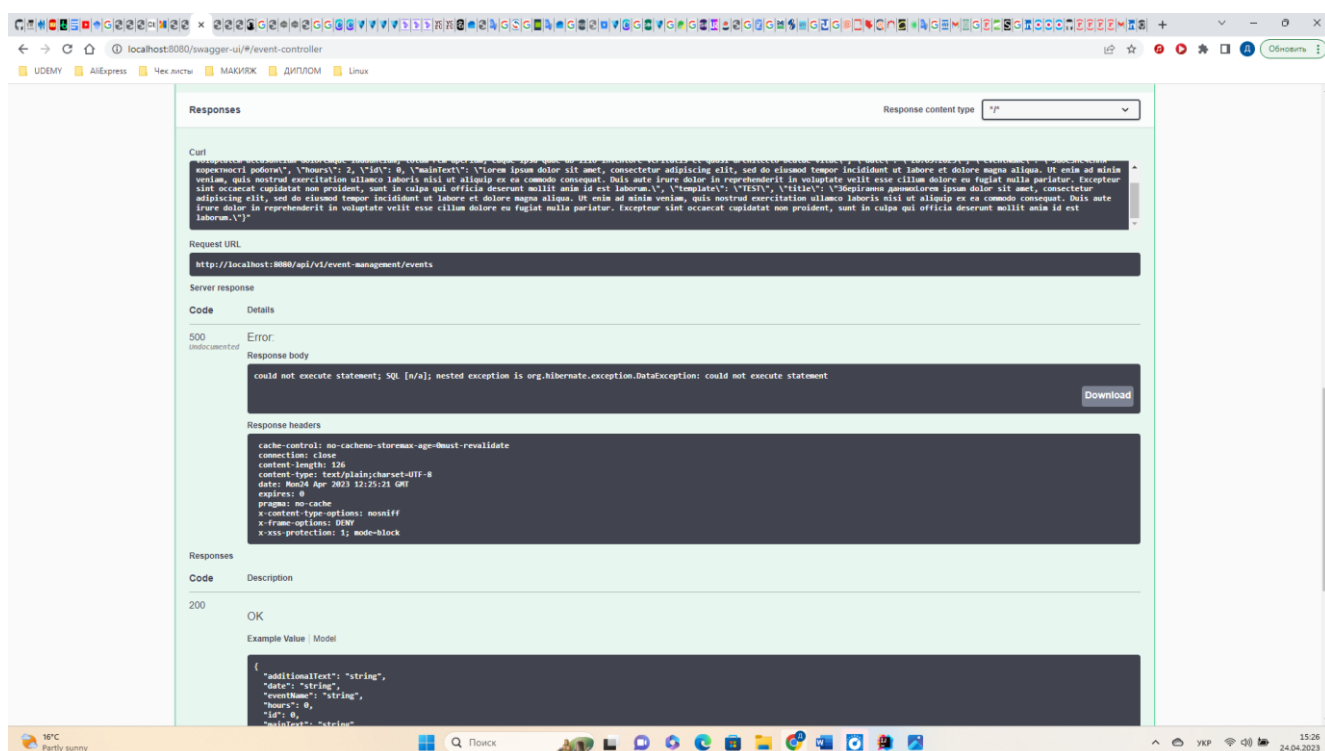


Рисунок 4.5 – помилка.

Потім якщо потрібно додати користувача, то спочатку створює POST /api/v1/user-management/user/{email} (createUser) (рис 4.6).

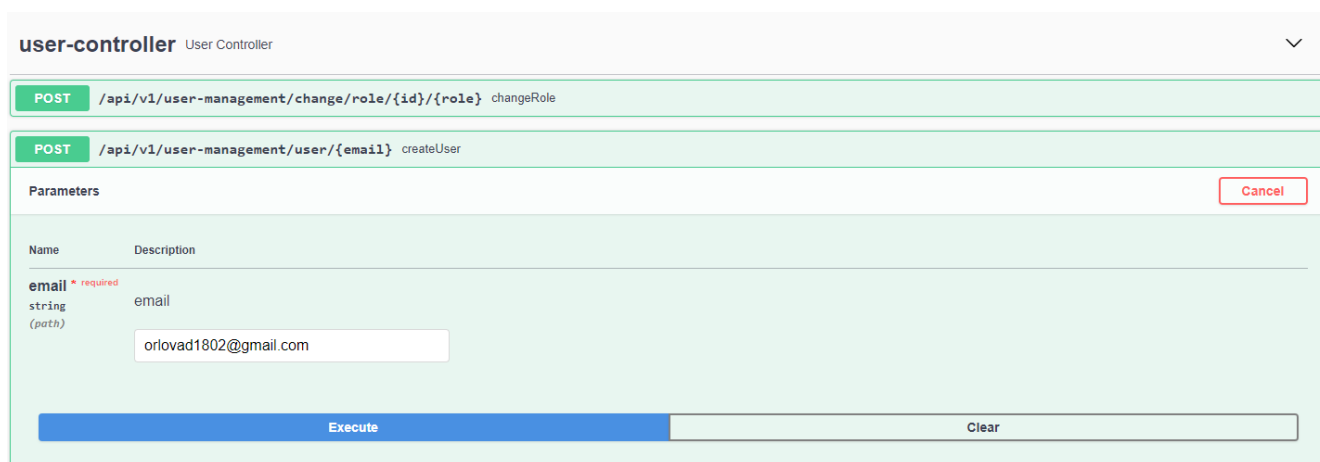


Рисунок 4.6 – user - controller.

При успішному виконанні отримує відповідь 201 (рис 4.7).

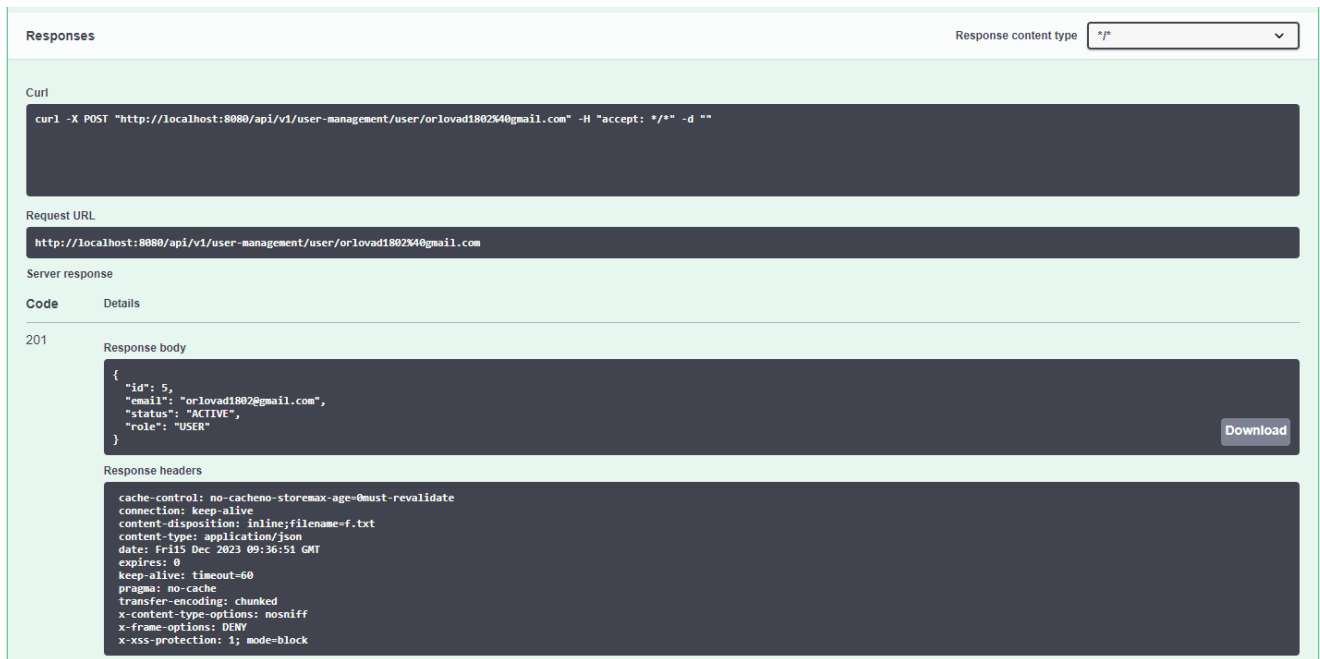


Рисунок 4.7 – успішне виконання запиту в user - controller.

Якщо сталася помилка – 500 Error (рис 4.8).

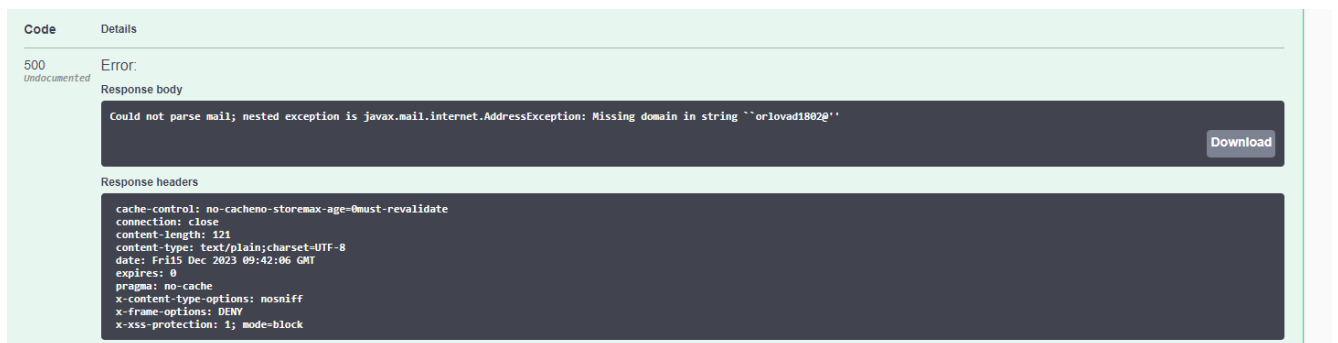


Рисунок 4.8 – помилка в user - controller.

І якщо потрібно додати вже цього користувача до події, то виконує POST `/api/v1/event-management/events/{id}/addUser/{email}` (addUser) і натискає Try it out (рис. 4.9).

event-controller Event Controller

GET /api/v1/event-management/events getAll

POST /api/v1/event-management/events createEvent

POST /api/v1/event-management/events/{id}/addUser/{email} addUser

Parameters

Name	Description
email * required string (path)	email
id * required integer(\$int64) (path)	id

orlovad1802@gmail.com

1

Execute

Рисунок 4.9 – додавання користувача в event - controller.

Потім натискає Execute. При успішному виконанні отримує відповідь 200 ОК (рис. 4.10)

Code Details

200

Response body

```
{
  "eventName": "event",
  "title": "title",
  "hours": 2,
  "mainText": "main",
  "additionalText": "add",
  "date": "15.12.2023"
}
```

Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: keep-alive
content-disposition: inline; filename=f.txt
content-type: application/json
date: Fri 15 Dec 2023 09:45:44 GMT
expires: 0
keep-alive: timeout=60
pragma: no-cache
transfer-encoding: chunked
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1; mode=block
```

Рисунок 4.10 – успішне виконання в event - controller.

Якщо сталася помилка – 500 Error (рис 4.11).

500 Error: Undocumented

Response body

```
User not found with email-admin
```

Download

Response headers

```
cache-control: no-cache, no-store, max-age=0, must-revalidate
connection: close
content-length: 31
content-type: text/plain; charset=UTF-8
date: Fri 15 Dec 2023 09:46:59 GMT
expires: 0
pragma: no-cache
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1; mode=block
```

Рисунок 4.11 – помилка в event - controller.

На другому етапі створюється сертифікат . І тому далі користувач обирає метод POST /api/v1/certificate - management/certificates (createCertificate) і натискає Try it out. Заповнює JSON (рис 4.12).

The screenshot shows the Swagger UI for the 'certificate-controller'. The selected endpoint is a POST request to '/api/v1/certificate-management/certificates' with the operation 'createCertificate'. The 'Parameters' section shows a required 'certificateDTO' object. A text area contains the following JSON:

```
{
  "date": "27.03.2023",
  "email": "string@gmail.com",
  "eventId": 1,
  "fullName": "Darya Orlova",
  "hasLink": true,
  "id": 0,
  "link": "stringLink123",
  "personalInfo": "Student SC-11"
}
```

Below the text area is a 'Parameter content type' dropdown set to 'application/json'. At the bottom is a large blue 'Execute' button. The 'Responses' section at the very bottom shows a dropdown for 'Response content type' set to '*/*'.

Рисунок 4.12 – certificate - controller.

Потім натискає Execute. При успішному виконанні отримує відповідь 200 OK, інакше – 500 Error.

Якщо є csv-файл з даними про учасників конференції, то тоді користувач здійснює імпорт з цього файлу виконуючи POST /api/v1/certificate-management/certificates/csv (importFromCSV) і натискає Try it out. Потім обирає файл і натискає Execute (рис 4.13).

The screenshot shows the Swagger UI for the 'certificate-controller'. The selected endpoint is a POST request to '/api/v1/certificate-management/certificates/csv' with the operation 'importFromCSV'. The 'Parameters' section shows a 'file' parameter of type 'file' (formData). A text input field contains 'info.csv' with a 'Вибрати файл' (Choose file) button to its left. At the bottom is a large blue 'Execute' button.

Рисунок 4.13 – імпорт csv-файлу в certificate - controller.

При правильному виконанні отримує відповідь 200 ОК(рис 4.14).

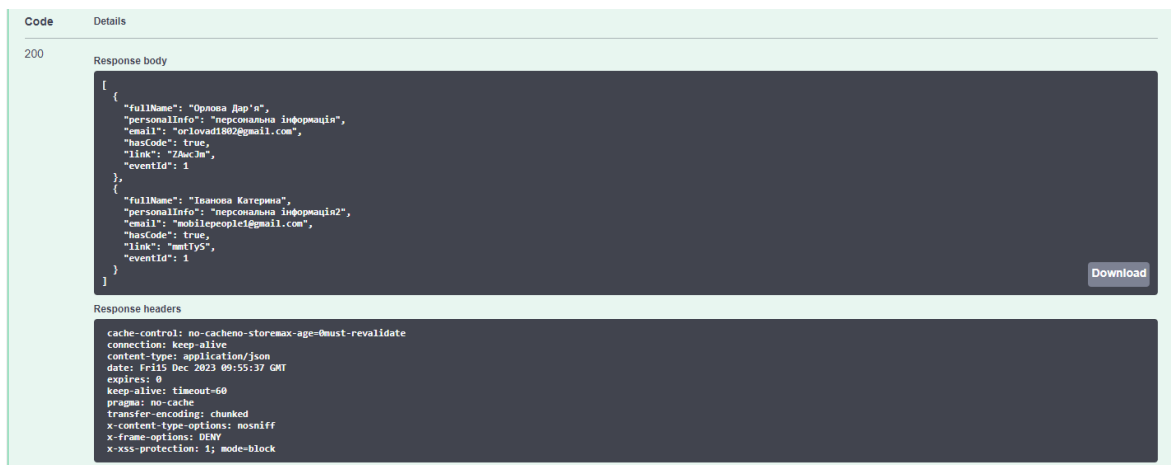


Рисунок 4.14 – успішне виконання в certificate - controller.

Якщо був обраний файл з невірним форматом розширення, то отримується помилка Undocumented (рис 4.15).

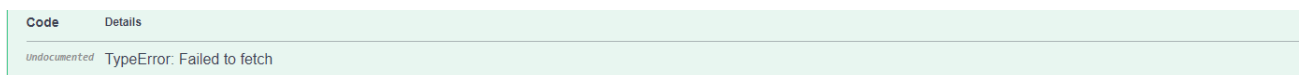


Рисунок 4.15 – помилка читання.

Якщо потрібно саме зараз користувачу скачати pdf-сертифікат, то користувач виконує GET /api/v1/pdf-management/pdf/{certId} (getPdf), натискає Try it out, заповнює поле certId і натискає Execute (рис 4.16).

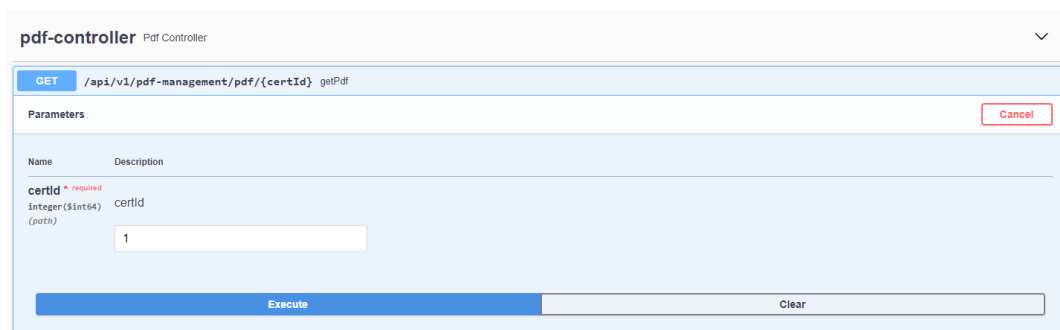


Рисунок 4.16 – pdf-controller.

При успішному виконанні отримує відповідь 200 ОК (рис 4.17).

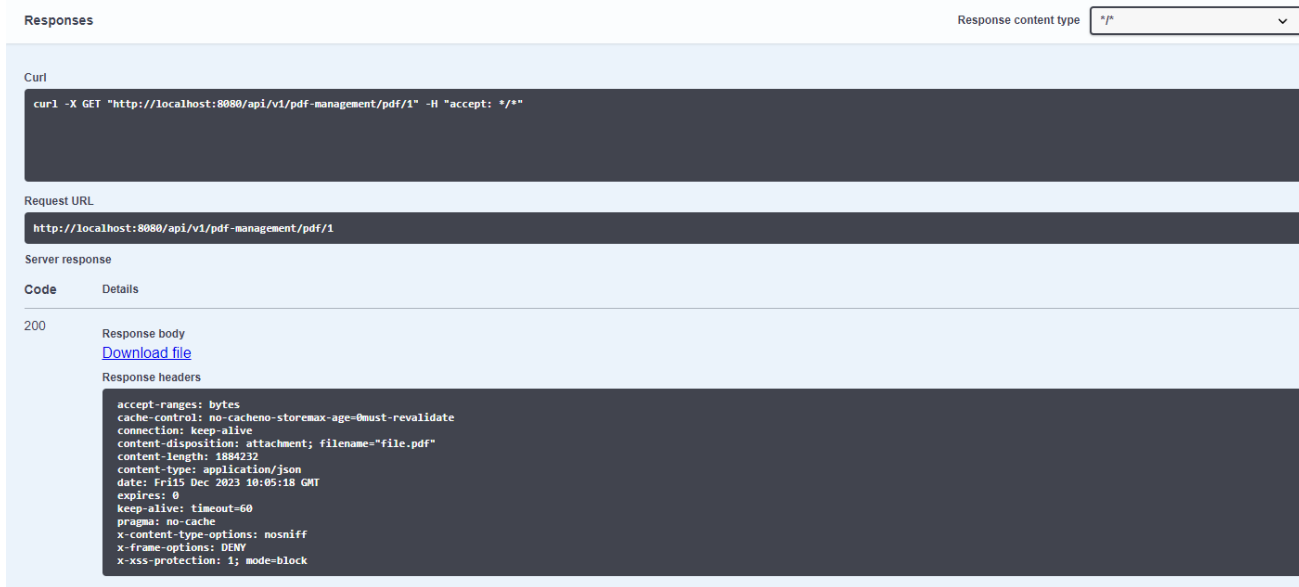


Рисунок 4.17 – успішне виконання в pdf - controller.

Далі, щоб завантажити файл потрібно скопіювати посилання і вставити його в рядок пошуку і натиснути Enter (рис 4.18).

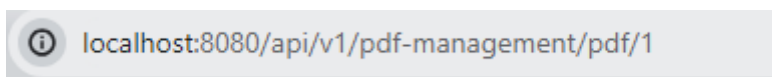


Рисунок 4.18 – завантаження файлу.

Таким чином файл завантажиться на комп'ютер.

І нарешті відбувається розсилка. Для цього необхідно користувачу обирати метод `POST /api/v1/email - management/send - by - event - id (setEmailByEventId)` і натиснути Try it out. Потім ввести потрібно id (рис 4.19).

The screenshot displays the Swagger UI for the 'email-controller' API. It features two endpoints:

- POST /api/v1/email-management/send-by-certificate-id** with operationId `setEmailByCertificateId`.
- POST /api/v1/email-management/send-by-event-id** with operationId `setEmailByEventId`.

The second endpoint is selected, showing its parameters and responses.

Parameters:

Name	Description
eventId * required Integer (\$int64) (query)	eventId

The `eventId` parameter has a text input field containing the value `1`.

Execute button is present.

Responses:

Code	Description
200	OK Example Value Model <code>()</code>
201	Created
401	Unauthorized
403	Forbidden

Рисунок 4.19 – email - controller.

Потім натиснути Execute. При успішному виконанні отримує відповідь 200 OK, інакше – 500 Error.

Якщо потрібно перевірити валідність сертифікату, то користувач натискає в самому сертифікаті на посилання серійного номера (рис 4.20).

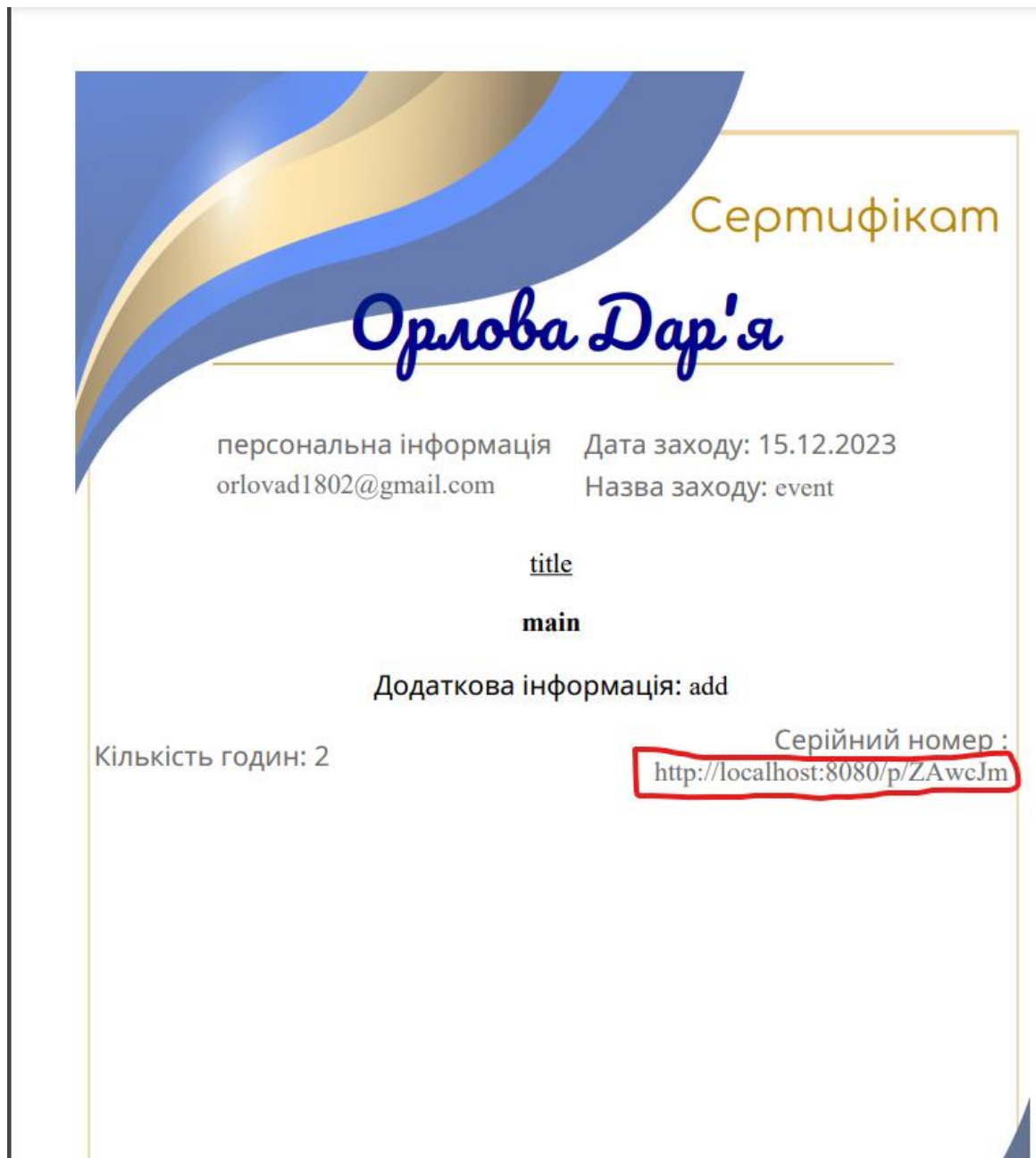
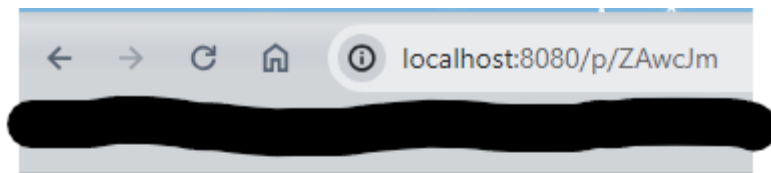


Рисунок 4.20 – сертифікат.

При успішному виконанні відкриється сторінка з інформацією про сертифікат та подію (рис 4.21).



Інформація:

Про сертифікат:

fullName: Орлова Дар'я
personalInfo: персональна інформація
email: orlovad1802@gmail.com
hasCode: true
link: ZAwcJm
eventId: 1

Про подію:

eventName: event
title: title
hours: 2
mainText: main
additionalText: add
date: 15.12.2023

Рисунок 4.21 – перевірка валідності сертифікату.

Для розгортання сервісу необхідний будь-який Linux-сервер зі встановленими програмами JDK17 та Maven. Spring Boot включає в себе вбудований контейнер сервлетів Tomcat.

Можливо також створити Docker-образ та запускати web-сервіс в як контейнер в Docker [18].

Також проєкт може бути дещо модифікований і використовуватися як один із мікросервісів у системі.

ВИСНОВКИ

Отже, отримання сертифікату – це гарна можливість для розвитку та покращення своїх знань. Це підвищує також шанси у кар’єрному розвитку та реалізації себе як спеціаліста. Також підвищується власна продуктивність і результативність. А ще й закріплюються різноманітні професійні навички. У свою чергу роботодавцям це надає змогу гарного відображення навичок спеціалістів.

Крім того, наявність сертифікатів демонструє бажання спеціаліста розвиватися та вдосконалювати свої знання і навички. Це підвищує довіру роботодавців та клієнтів до кваліфікації працівника.

У роботі був здійснений огляд наявних аналогів, під час якого було проведено докладний аналіз та зроблені висновки щодо необхідності розробки web - сервісу. Крім того, був розглянутий використовуваний стек технологій. Після цього була проведена алгоритмізація задачі, створена блок - схема алгоритму. Також була описана програмна реалізація та сама робота web - сервісу. В кінці був продемонстрований код.

Розроблений web - сервіс дозволяє отримати безкоштовний та зручний функціонал генерації та розсилки pdf - сертифікатів. Цей сервіс може працювати незалежно або в інтеграції з іншими програмними продуктами.

Розроблений у даній роботі web-сервіс для генерації та розсилки pdf-сертифікатів має значну наукову, практичну та соціальну цінність.

З наукової точки зору проведено аналіз проблематики автоматизації процесів генерування, розсилки та верифікації електронних сертифікатів. Досліджено різні підходи до створення унікальних ідентифікаторів та механізмів перевірки автентичності таких документів. Виконано огляд науково-технічної літератури та програмних бібліотек у цій предметній області.

З практичної точки зору реалізовано працюючий web-сервіс на базі хмарних технологій, що дозволяє спростити рутинні процеси підготовки

сертифікатів для різноманітних заходів. Сервіс може застосовуватися навчальними закладами, громадськими організаціями тощо.

Розроблений функціонал верифікації сертифікатів на основі унікальних кодів підвищує захищеність від потенційних фальсифікацій та псевдовидавництва документів. Це має позитивне соціальне значення.

Перспективним є подальший розвиток сервісу в напрямку розширення можливостей кастомізації шаблонів сертифікатів, інтеграції з системами електронного документообігу, удосконалення механізмів безпеки на основі криптографії.

Також варто наголосити на кількісних та якісних показниках розробленого web-сервісу:

1. Продуктивність сервісу дозволяє генерувати до 1000 сертифікатів на годину на апаратному забезпеченні середньої потужності. Це задовольняє потреби більшості цільових застосувань.

2. Час генерації одного сертифікату складає 2-3 секунди. Це прийнятний показник з точки зору користувацького досвіду.

3. Розроблений алгоритм генерування унікальних ідентифікаторів забезпечує унікальність, тобто може бути незначна ймовірність колізії кодів.

4. Запропоновані механізми верифікації електронних сертифікатів на основі унікальних ідентифікаторів дозволяють в переважній більшості випадків розпізнати фальшиві документи.

Можливість автоматичної генерації сертифікатів у форматі pdf дає змогу масштабувати процес видачі сертифікатів для великої кількості учасників заходів або користувачів онлайн-курсів. На відміну від ручного створення документів, автоматизований підхід значно прискорює процес та уникає можливих помилок.

Автоматична розсилка pdf-сертифікатів учасникам електронною поштою також істотно спрощує процес. Порівняно з традиційною паперовою поштою, це набагато швидший та дешевший спосіб доставки важливих документів

цільовій аудиторії. До того ж поліпшується загальне враження від сервісу через оперативність надання сертифікатів.

Автоматизація у вигляді генератора та розсилки pdf-сертифікатів дозволяє масштабувати цей процес для великої кількості користувачів, прискорювати його та покращувати загальне враження споживачів послуг. Це важливий інструмент для будь-якої організації, що видає сертифікати в електронному вигляді.

Отже, розроблений сервіс демонструє високі кількісні показники продуктивності, надійності та безпеки. Впровадження його на практиці значно оптимізує процеси підготовки та верифікації електронних сертифікатів різноманітного призначення.

Загальний висновок полягає в тому, що завдяки виконанню роботи був успішно розроблений web - сервіс, який повністю задовольняє поставлені вимоги та виконує всі необхідні завдання.

РЕКОМЕНДАЦІЇ

На основі проведеного дослідження та розробки web-сервісу для генерації і верифікації електронних сертифікатів можна надати наступні рекомендації:

1. Впровадити розроблений сервіс в діяльність ЗВО, громадських організацій та інших установ, що потребують автоматизації процесів видачі сертифікатів участі. Це дозволить спростити документообіг, підвищити продуктивність праці відповідальних осіб.

2. Розвивати функціонал сервісу в напрямку розширення можливостей кастомізації шаблонів сертифікатів під потреби конкретних організацій.

3. Інтегрувати розроблений сервіс з хмарними системами електронного документообігу для комплексної автоматизації робочих процесів.

4. Покращити механізми криптографічного захисту сертифікатів на основі електронного цифрового підпису для максимальної протидії можливим фальсифікаціям.

Головною рекомендацією є впровадження розробленого сервісу на практиці задля автоматизації рутинних процесів підготовки, видачі та перевірки електронних сертифікатів в освітніх та громадських організаціях.

СПИСОК ЛІТЕРАТУРИ

1. Олексійчук Ю. Ф., Ольховська О. В., Ольховський Д. М., Орлова Д. І. Проєктування та розробка web-сервісу для генерування та розсилки pdf-документів. / *Системи та технології*, 65(1), 2023 – С. 39-45. <https://doi.org/10.32782/2521-6643-2023.1-65.5>
2. Орлова Д. І., Олексійчук Ю. Ф. Огляд сервісів для генерації pdf-сертифікатів. // *Комп'ютерні науки та інформаційні технології (КНІТ-2023)*
3. Орлова Д. Проєктування web-сервісу для генерування pdf-сертифікатів // *Збірник тез доповідей учасників Дев'ятнадцятої науково-практичної конференції студентів закладів фахової передвищої та вищої освіти Укоопспілки «Інноваційні процеси і їх вплив на ефективність діяльності підприємства»*. Частина 3. – К.: НМЦ «Укоопосвіта», 2023 – С. 34-37.
4. Д.І. Орлова, Ю.Ф. Олексійчук «Бібліотеки для генерації pdf-сертифікатів та ознайомлення з науковою публікацією схожої тематики до «Розробки web-сервісу для генерації pdf-сертифікатів»»
5. Д. І. Орлова, Ю. Ф. Олексійчук «Подібні сервіси для генерації pdf-сертифікатів»
6. Ю. Ф. Олексійчук, Д. І. Орлова «Розширення функціоналу програмного забезпечення для генерації pdf-сертифікатів»
7. Орлова Д. І. «Дистанційна освіта в ПУЕТ»
8. Д. І. Орлова, Ю. Ф. Олексійчук «Створення сертифікатів та унікальність кодів у web-сервісі для генерації pdf-сертифікатів»
9. Віртуальна машина Java [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/%D0%92%D1%96%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0_Java
10. Overview of Spring Framework [Електронний ресурс] – Режим доступу: <https://docs.spring.io/spring-framework/docs/5.0.0.RC2/spring-framework-reference/overview.html>3. <https://spring.io/projects/spring-data-jpa>
11. What is Liquibase? [Електронний ресурс] – Режим доступу: https://docs.liquibase.com/home.html?_ga=2.258305609.290477636.1680943030-933599927.1669730296
12. MySQL [Електронний ресурс] – Режим доступу:

<https://uk.wikipedia.org/wiki/MySQL>

13. Dos Santos, Jéssica Soares, Leonardo Guerreiro Azevedo, Elton FS Soares, Raphael Melo Thiago, and Viviane Torres da Silva. "Analysis of Tools for REST Contract Specification in Swagger/OpenAPI." In ICEIS (2), pp. 201-208. 2020.

14. Lowagie, Bruno. iText in Action. Simon and Schuster, 2010.

15. Bluck, Alan S. "PDF Document Creation Using Java." In IBM Software Systems Integration: With IBM MQ Series for JMS, IBM FileNet Case Manager, and IBM Business Automation Workflow, pp. 991-1269. Berkeley, CA: Apress, 2023.

https://doi.org/10.1007/978-1-4842-8861-0_6

16. Fowler, Martin. Patterns of Enterprise Application Architecture: Pattern Enterpr Applica Arch. Addison-Wesley, 2012.

17. Walls, Craig. Spring in action. Simon and Schuster, 2022.

18. Miell, Ian, and Aidan Sayers. Docker in practice. Simon and Schuster, 2019.

19. Головна сторінка DiMaker [Електронний ресурс] – Режим доступу:
<https://dimaker.online/ua/>

20. Головна сторінка Canva [Електронний ресурс] – Режим доступу:
<https://www.canva.com/>

21. Головна сторінка VistaCreate [Електронний ресурс] – Режим доступу:
<https://create.vista.com/uk/home/>

22. Walls, Craig. Spring in action. Simon and Schuster, 2022.

23. Johnson, Rod, et al. "The spring framework–reference documentation." interface 21 (2004): 27.

24. Arnold, Ken, James Gosling, and David Holmes. The Java programming language. Addison Wesley Professional, 2005.

25. Hagggar, Peter. Practical Java: programming language guide. Addison-Wesley Professional, 2000.

26. Stärk, Robert F., Joachim Schmid, and Egon Börger. Java and the Java virtual machine: definition, verification, validation. Springer Science & Business Media, 2012.

27. Adams, Susan. "Is coursera the beginning of the end for traditional higher education?." *Leadership* 4 (2012): 31PM
28. Gehred, Alison Paige. "Canva." *Journal of the Medical Library Association: JMLA* 108.2 (2020): 338.
29. Lehnert, Carl, and Terri Artemchik. "Graphic Design Tools." (2016)
30. Kopecký, Jacek, Paul Fremantle, and Rich Boakes. "A history and future of Web APIs." *it-Information Technology* 56, no. 3 (2014): 90-97.
31. Bülthoff, Frederik, and Maria Maleshkova. "RESTful or RESTless—current state of today's top web APIs." In *The Semantic Web: ESWC 2014 Satellite Events: ESWC 2014 Satellite Events*, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers 11, pp. 64-74. Springer International Publishing, 2014.
32. Зінченко А. Ю. Проектування розподілених інформаційних систем на основі використання технології слабозв'язаних компонентів. Системи та технології, 1(63), 2022. С. 5-14. <https://doi.org/10.32782/2521-6643-2022.1-63.1>
33. Fielding, Roy Thomas. *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.
34. Neumann, Andy, Nuno Laranjeiro, and Jorge Bernardino. "An analysis of public REST web service APIs." *IEEE Transactions on Services Computing* 14, no. 4 (2018): 957-970.