

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

«Розробка інформаційної системи оптичного розпізнавання тексту й екстракції слабоструктурованих даних»

Студента 2м курсу, 6 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

**Дьяченко Максима
Ігоровича**

Науковий керівник
асистент кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

**Гнатченко Дмитро
Дмитрович**

Гарант освітньої програми
доктор технічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис гаранта

**Криворучко Олена
Володимирівна**

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«8» листопада 2019 р.

Завдання на випускний кваліфікаційний проект студентів

Дьяченко Максиму Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту «Розробка інформаційної системи оптичного розпізнавання тексту й екстракції слабоструктурованих даних»

Затверджена наказом ректора від «12» грудня 2019 р. № 4304

2. Строк здачі студентом закінченої проекту 01 грудня 2020 р.

3. Цільова установка та вихідні дані до проекту

Мета проекту є розробка інформаційної системи з використанням технологій оптичного розпізнавання символів та екстракції слабоструктурованих даних.

Об'єкт дослідження є технології оптичного розпізнавання символів та екстракції слабоструктурованих даних

Предмет дослідження є інструменти що використовується у процесі проектування та розробки та розробки інформаційних систем.

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ РОЗПІЗНАВАННЯ ТЕКСТУ ТА ЕКСТРАКЦІЇ ДАНИХ І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ЯКОМУ ВОНИ ВИКОРИСТОВУЮТЬСЯ

1.1. Принципи роботи технологій оптичного розпізнавання символів

1.2. Принципи роботи технологій екстракції слабоструктурованих даних

1.3. Область використання технологій та огляд існуючих програмних рішень в яких вони використовуються.

1.4. Висновок до розділу 1

РОЗДІЛ 2. ХАРАКТЕРИСТИКА ІНФОРМАЦІЙНОЇ СИСТЕМИ, ПРОЕКТУВАННЯ БАЗИ ДАНИХ ТА ІНТЕРФЕЙСУ КОРИСТУВАЧА

2.1. Опис процедури виконання якої забезпечить інформаційна система

2.2. Етапи проектування бази даних

2.3. Етапи проектування інтерфейсу користувача

2.5. Висновок до розділу 2

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЮ УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ СИСТЕМОЮ

3.1. Опис програмного модулю з управління інформаційною системою та коду її інтерфейсу

3.2. Опис модулю розпізнавання тексту та екстракції даних

3.3. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання проекту

№ пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів проекту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проекту</i>	20.09.2019	20.09.2019
2.	<i>Розробка та затвердження завдання на проект магістра</i>	08.11.2019	08.11.2019
3.	<i>Вступ та перелік літературних джерел</i>	24.01.2020	24.01.2020
4.	<i>Наукова стаття</i>	01.09.2020	01.09.2020
5.	<i>Технічне завдання</i>	28.02.2020	28.02.2020
6.	<i>Розділ 1. Дослідження технологій розпізнавання тексту та екстракції даних</i>	25.06.2020	25.06.2020
7.	<i>Розділ 2. Характеристика інформаційної системи, проектування бази даних та інтерфейсу користувача</i>	07.09.2020	07.09.2020
8.	<i>Розділ 3. Розробка програмного модулю управління інформаційною системою</i>	19.10.2020	19.10.2020
9.	<i>Програма та методика тестування</i>	21.10.2020	21.10.2020
10.	<i>Керівництво користувача</i>	23.10.2020	23.10.2020
11.	<i>Висновки та пропозиції</i>	30.10.2020	30.10.2020
12.	<i>Здача випускного кваліфікаційного проекту на кафедрі (перша перевірка)</i>	05.11.2020	05.11.2020
13.	<i>Підготовка автореферату та презентації доповіді</i>	05.11.2020	05.11.2020
14.	<i>Попередній захист випускного кваліфікаційного проекту</i>	25.11.2020- 27.11.2020	25.11.2020 -27.11.2020
15.	<i>Зовнішнє рецензування випускного кваліфікаційного проекту</i>	01.12.2020	01.12.2020
16.	<i>Підготовка до публічного захисту випускного кваліфікаційного проекту</i>	8.12.2020- 9.12.2020	

7. Дата видачі завдання «8» листопада 2019 р.

8. Науковий керівник випускного кваліфікаційного проекту Гнатченко Д. Д.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Криворучко О.В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Дьяченко М.І.

(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

Відмітка про попередній захист _____
(ПІБ, підпис, дата)

Випускний кваліфікаційний проект студента Дьяченко М. І.
(прізвище, ініціали)
може бути допущена до захисту екзаменаційній комісії.

Криворучко О. В.
(прізвище, ініціали, підпис)
Криворучко О. В.
(підпис, прізвище, ініціали)

Криворучко О. В.
(підпис, прізвище, ініціали)

« _____ » 20 ____ г.

АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці інформаційної системи із використанням технологій оптичного розпізнавання символів та екстракції слабоструктурованих даних для автоматизації процесу валідації та управління лабораторними роботами у освітніх закладах.

В результаті було створено таку інформаційну систему.

Розробка програмної частини виконана у текстовому редакторі Sublime Text 3. Обрана мова програмування – PHP. Окремі елементи системи розроблені із використанням HTML, SQL, JSON файлів відповідно до стандартів розробки веб-додатків. Модуль розпізнавання тексту та екстракції даних був розроблений як проект у програмному забезпеченні ABBYY FlexiCapture. Запуск проекту виконувався на сервері Apache що входить до складу програмного забезпечення MAMP для MacOS.

Готовий програмний комплекс EasyLab було успішно протестовано відповідно до функціональних вимог.

Ключові слова: PHP, веб-додаток, OCR, екстракція даних, інформаційна система.

ABSTRACT

According to the purpose the research is devoted to developing an information system using optical character recognition and data capture technologies for process automation in management and validation of laboratory exercises in educational institutions.

As a result, such system was developed.

The server-side development was done in text editor Sublime Text 3. The chosen programming language is PHP. Separate elements of the mobile application were designed using HTML, SQL, JSON files in accordance with the standards of

web applications development. Optical character recognition and data extraction module was developed using ABBYY FlexiCapture software. Project was deployed on the Apache web server that is part of MAMP pack for MacOS.

The EasyLab system was successfully tested according to functional requirements.

Keywords: PHP, web application, OCR, data capture, information system.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

ОС – операційна система

СУБД – система управління базами даних

OCR (англ. Optical Character Recognition) – оптичне розпізнавання символів

API (англ. Application Programming Interface) – програмний інтерфейс додатку

JSON (англ. JavaScript Object Notation) – текстовий формат обміну даних

MAMP (англ. MacOS Apache MySQL PHP) – набір із засобів розробки, утиліт і документації до певної технології або платформи

MD5 (англ. Message Digest 5) — 128-бітний алгоритм хешування

					<i>КНТЕУ 121–06-08.МР</i>			
					Розробка інформаційної системи оптичного розпізнавання тексту й екстракції слабоструктурованих даних	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		В	3	52
Зав. каф.		Криворучко О.В.			Зміст	Факультет інформаційних технологій, 2 курс, 6 група		
Керівник		Гнатченко Д.Д.						
Гарант		Криворучко О.В.						
Розробив		Дьяченко М.І.						

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ РОЗПІЗНАВАННЯ ТЕКСТУ ТА ЕКСТРАКЦІЇ ДАНИХ І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ЯКОМУ ВОНИ ВИКОРИСТОВУЮТЬСЯ	7
1.1. Принципи роботи технологій оптичного розпізнавання символів	7
1.2. Принципи роботи технологій екстракції слабоструктурованих даних	11
1.3. Область використання технологій та огляд існуючих програмних рішень в яких вони використовуються	16
1.4. Висновки до розділу 1	21
РОЗДІЛ 2. ХАРАКТЕРИСТИКА ІНФОРМАЦІЙНОЇ СИСТЕМИ, ПРОЕКТУВАННЯ БАЗИ ДАНИХ ТА ІНТЕРФЕЙСУ КОРИСТУВАЧА	23
2.1. Опис процедури виконання якої забезпечить інформаційна система	23
2.2. Етапи проектування бази даних.....	24
2.3. Етапи проектування інтерфейсу користувача	29
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО МОДУЛЮ УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ СИСТЕМОЮ	38
3.1. Опис програмного модулю з управління інформаційною системою та коду її інтерфейсу.....	38
3.2. Опис модулю розпізнавання тексту та екстракції даних	44

					<i>КНТЕУ 121– 06-08.МР</i>			
					Розробка інформаційної системи оптичного розпізнавання тексту й екстракції слабоструктурованих даних	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		В	4	52
Зав. каф.		Криворучко О.В.						
Керівник		Гнатченко Д.Д.						
Гарант		Криворучко О.В.						
Розробив		Дьяченко М.І.			<i>Зміст</i>	Факультет інформаційних технологій, 2 курс, 6 група		

3.3. Висновки до розділу 3	48
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50
ТЕХНІЧНЕ ЗАВДАННЯ.....	51

					<i>КНТЕУ 121 06-08.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		5

ВСТУП

Сьогодні у всьому світі поширюється використання інформаційних технологій у різних сферах діяльності людини. Разом з цим зменшується кількість інформації, що зберігається на папері, і все більша кількість організацій переходять до електронного документообігу для збільшення швидкості і якості обробки інформації. Одним із актуальних питань процесу переходу є розробка технологій і програмного забезпечення екстракції даних із слабоструктурованих документів. Розробка інформаційної системи з використанням цих технологій є актуальною задачею для багатьох зарубіжних і вітчизняних організацій.

Дані технології можуть та вже використовуються у багатьох сферах діяльності людини, таких як медицина, фінанси, державне управління, освіта та інші. Перспективи, що розкриваються із залученням цих технологій у внутрішні процеси організації, включають в себе пришвидшення процесів обробки даних, збереження коштів, оптимізація та підвищення ефективності, адже для певного процесу, в якому було залучено певну кількість людей, можна створити систему, що виконуватиме цю роботу ефективніше.

Метою дослідження є розробка інформаційної системи з використанням технологій оптичного розпізнавання символів та екстракції слабоструктурованих даних.

Об'єктом дослідження є технології оптичного розпізнавання символів та екстракції слабоструктурованих даних.

					<i>КНТЕУ 121– 06-08.МР</i>			
					Розробка інформаційної системи оптичного розпізнавання тексту й екстракції слабоструктурованих даних	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
						В	6	52
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>Вступ</i>		
Зав. каф.		Криворучко О.В.						
Керівник		Гнатченко Д.Д.						
Гарант		Криворучко О.В.						
Розробив		Дьяченко М.І.			Факультет інформаційних технологій, 2 курс, 6 група			

Предметом дослідження є інструменти що використовується у процесі проектування та розробки та розробки інформаційних систем.

Задачі дослідження:

- Проаналізувати можливості та перспективи використання технологій;
- Розробити інформаційну систему із використанням даних технологій;
- Освоїти інструменти необхідні для імплементації існуючих технологій;

Наукова новизна дослідження полягає у застосуванні технологій з метою валідації документів у сфері освіти.

					<i>КНТЕУ 121 06-08.MP</i>	Аркуш
						7
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 1

ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ РОЗПІЗНАВАННЯ ТЕКСТУ ТА ЕКСТРАКЦІЇ ДАНИХ І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ЯКОМУ ВОНИ ВИКОРИСТОВУЮТЬСЯ

1.1. Принципи роботи технологій оптичного розпізнавання символів

OCR (Optical Character Recognition – оптичне розпізнавання символів) - це технологія, яка перетворює зображення, що містять текст, у формати з текстом, який можна редагувати. OCR дозволяє обробляти відскановані книги, знімки екрана та фотографії з текстом та отримувати такі документи, як файли TXT, DOC або PDF. Ця технологія широко використовується у багатьох сферах. Найдосконаліші системи OCR можуть обробляти майже будь-які типи зображень, навіть такі складні, як скановані сторінки журналів із зображеннями та колонками, або фотографії з мобільного телефону.

Процес перетворення зображення в документ, який можна редагувати, розділений на кілька етапів[1]. Кожен крок - це набір пов'язаних алгоритмів, які виконують частину роботи OCR. Загальні етапи процесу OCR такі:

1. Завантаження растрового зображення із заданого джерела. Джерелом може бути файл або вказівник на пристрій вводу. Більше того, хороша система OCR повинна розуміти багато форматів зображень: BMP, TIFF (як односторінкові, так і багатосторінкові зображення), JPEG, PNG тощо. Він також повинен підтримувати файли PDF, оскільки багато документів

					<i>КНТЕУ 121 06-08.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			25.06.20	Розробка інформаційної системи оптичного розпізнавання тексту й екстракції слабоструктурованих даних	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Гнатченко Д.Д			25.06.20		<i>РІ</i>	<i>8</i>	<i>52</i>
Гарант	Криворучко О.В.			25.06.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив	Дьяченко М.І.			25.06.20				
					<i>Дослідження технологій оптичного розпізнавання тексту та екстракції слабоструктурованих даних</i>			

зберігаються як зображення у форматі PDF, і єдиним способом вилучення тексту з таких файлів є виконання OCR.

2. Виявлення найважливіших характеристик зображення, таких як роздільна здатність та інверсія. Багато алгоритмів OCR очікують заздалегідь визначеного діапазону розмірів шрифту та кольорів переднього / фонового зображень, тому перед обробкою зображення потрібно обробити масштабом та інвертувати, якщо це необхідно.

3. Зображення може бути перекошене, або воно може містити багато шуму, тому для поліпшення якості зображення застосовуються алгоритми вирівнювання та шумозаглушення.



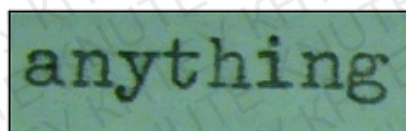
Original image



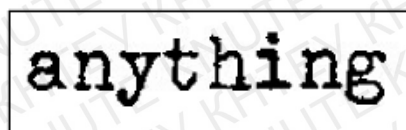
Image after Deskw algorithm

Рис. 1.1. Застосування алгоритмів вирівнювання та шумозаглушення

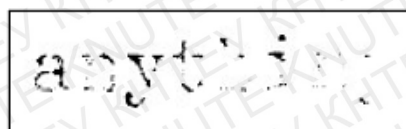
4. Багато алгоритмів OCR можуть обробляти лише бінаризовані зображення, тому кольорові або сірі зображення повинні бути перетворені в бінаризовані. Процес називається "бінаризація". Цей крок дуже важливий, оскільки неправильна бінаризація може спричинити масу проблем із подальшим розпізнаванням.



Original image



Correct binarization



Incorrect binarization

Рис. 1.2. Застосування алгоритмів бінаризації

5. Виявлення та видалення ліній. Цей крок необхідний для вдосконалення аналізу макета сторінки, досягнення кращої якості розпізнавання підкресленого тексту, виявлення таблиць тощо.

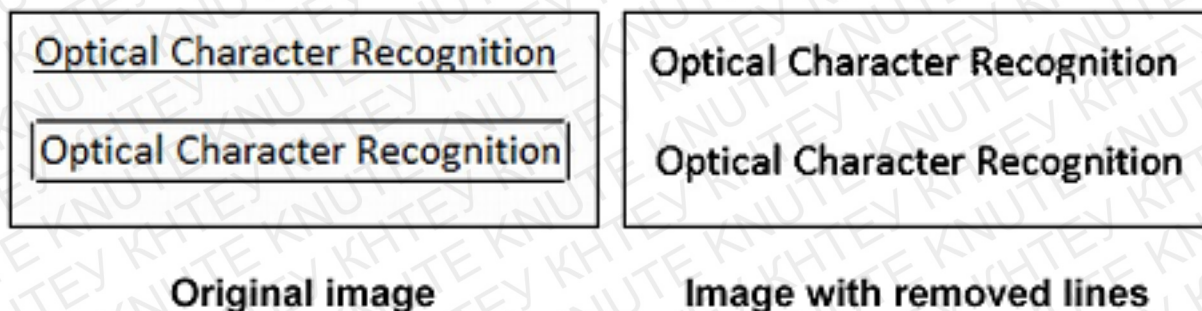


Рис. 1.3. Виявлення та видалення ліній

6. Аналіз макета сторінки (також званий "зонуванням"). Система OCR повинна визначати положення та типи всіх важливих областей зображення.

7. Виявлення рядків тексту та слів. Іноді це непросте завдання через різний розмір шрифту та малі пробіли між словами.

					КНТЕУ 121 06-08.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		10

8. Комбіновано-розбитий аналіз символів. Часто деякі символи розбиваються на кілька частин, або деякі символи торкаються один одного. Необхідно виявити такі випадки і визначити правильне положення кожного символу.

9. Розпізнавання символів. Це основний алгоритм OCR. Зображення кожного символу має бути перетворено у відповідний код символу. Іноді цей алгоритм створює кілька символічних кодів для невизначених зображень. Наприклад, розпізнавання зображення символу "I" може створити коди для "I", "|", "1", "l"; остаточний код символу буде обраний пізніше.

10. Підтримка словника. Цей крок може покращити якість розпізнавання. Деякі символи, такі як "1" та "l", "C" та "G", можуть виглядати дуже схоже, і словник може допомогти прийняти рішення.

11. Збереження результатів у вибраному вихідному форматі, наприклад, PDF, DOC, RTF або TXT з можливістю пошуку. Важливо зберегти оригінальний макет сторінки: стовпці, шрифти, кольори, картинки, фон тощо.

Вище наведено спрощений алгоритм. Багато інших незначних алгоритмів також повинні бути реалізовані для досягнення хорошого розпізнавання на різних типах зображень, але вони в більшості випадків не є головними і можуть відрізнятися в різних системах OCR.

Кожен крок OCR дуже важливий. Весь процес OCR провалиться, якщо будь-який крок не зможе правильно обробити дане зображення. Кожен алгоритм повинен функціонувати на найбільшому діапазоні зображень, тому на ринку існує лише кілька хороших універсальних систем OCR. З іншого боку, якщо відомі деякі характеристики даних зображень, завдання стає набагато простішим. Ви можете отримати кращу якість розпізнавання, якщо потрібно обробити лише один тип зображень. Для досягнення найкращих результатів, якщо відомі деякі особливості зображень, хороша система OCR повинна мати можливість регулювати найважливіші параметри кожного

					КНТЕУ 121 06-08.МР	Аркуш
						11
Зм.	Аркуш	№ докум	Підпис	Дата		

алгоритму. Іноді це єдиний спосіб поліпшити якість розпізнавання. Зараз не існує систем OCR, які б були настільки хорошими, як люди, і тому ми не можемо очікувати від системи OCR якості в 100%. Розпізнавання написаного на машинці тексту латинським шрифтом все ще не є на 100% точним навіть там, де доступні чіткі зображення. Одне дослідження[2], засноване на розпізнаванні сторінок газет 19-го і початку 20-го століття, прийшло до висновку, що точність розпізнавання OCR для комерційного програмного забезпечення OCR варіюється від 81% до 99%.

1.2. Принципи роботи технологій екстракції слабоструктурованих даних

Неструктуровані дані - це інформація, яка або не має заздалегідь визначеної моделі даних, або не організована заздалегідь визначеним чином. Неструктурована інформація, як правило, містить багато тексту, але може містити такі дані, як дати, цифри та факти. Це призводить до порушень та неоднозначностей, які ускладнюють розуміння використання традиційних програм порівняно з даними, що зберігаються у структурованих форматах у базах даних або семантично позначені у документах.

Програмне забезпечення, яке створює машинно оброблювану структуру, може використовувати лінгвістичну, слухову та візуальну структуру, що існує у всіх формах людського спілкування. Алгоритми можуть зробити висновок про цю властиву структуру з тексту, наприклад, вивчаючи морфологію слів, синтаксис речень та інші характеристики документів. Потім неструктурована інформація може бути збагачена і позначена для вирішення двозначностей та реалізації методів, що базуються на релевантності, а потім використана для полегшення пошуку та виявлення важливої інформації. Приклади "неструктурованих даних" можуть включати книги, журнали, документи,

					КНТЕУ 121 06-08.МР	Аркуш
						12
Зм.	Аркуш	№ докум	Підпис	Дата		

метадані, медичні записи, аудіо, відео, аналогові дані, зображення, файли та неструктурований текст, такий як тіло повідомлення електронної пошти, веб-сторінка тощо. Основний вміст, що передається, не має визначеної структури, він, як правило, упаковується в об'єкти (наприклад, у файли або документи), які самі мають структуру і, таким чином, є сумішшю структурованих та неструктурованих даних, але в сукупності це все ще називають "слабоструктурованими" даними. Наприклад, веб-сторінка HTML позначається тегами, але розмітка HTML зазвичай служить виключно для візуалізації. Вона не фіксує значення або функції позначених елементів у спосіб, що підтримує автоматизовану обробку інформаційного вмісту сторінки. Теги XHTML дозволяють машинну обробку елементів, хоча, як правило, не фіксують і не передають семантичного значення позначених термінів.

У сучасному програмному забезпеченні процес екстракції даних зазвичай поділяється на п'ять етапів кожен з яких, в свою чергу, може бути налаштований або кастомізований для того, щоб задовольнити потреби конкретної можелі обробки даних. Ці етапи:

- Імпорт
- Розпізнавання
- Застування алгоритмів екстракції даних
- Перевірка (верифікація)
- Експорт

Перший етап імпорту може бути виконаний як для паперових документів із використанням сканерів, фотоапаратів, так і для електронних документів різними способами наприклад автопатичний імпорт документів з електронної пошти, викоритання API (Application Programming Interface), або просто постійна перевірка локальної папки на машині в очікування появи у ній файлу для подальшої обробки. Також, до стадії імпорту можна віднести і

					КНТЕУ 121 06-08.MP	Аркуш
						13
Зм.	Аркуш	№ докум	Підпис	Дата		

попередню обробку зображень, оскільки у багатьох випадках виконання певних маніпуляцій із зображеннями може покращити якість подальшого розпізнавання. Методи передобробки були детальніше розглянуті в підрозділі 1.1.

Етап розпізнавання, окрім використання вже розглянутої технології OCR може також включати використання технологій[3,4]:

- OMR (Optical Mark Recognition). Технологія OMR дозволяє користувачеві читати прапорці, тобто знаки, проставлений у заздалегідь визначених пробілах. Дана технологія широко використовується у автоматизації процесу перевірки тестів, наприклад ЗНО.



Рис. 1.3. Поля для розпізнавання технологією OMR

- BCR (Bar Code Recognition). Технологія BCR дозволяє розпізнавати штрих-коди, декодуючи їх вміст. Часто розпізнавання вмісту штрих-кодів може допомогти класифікувати вхідні документи і додати точності до результатів екстракції.
- ICR (Intelligent Character Recognition). За допомогою технології ICR можна розпізнавати рукописний текст, коли між символами є простір. Дана технологія використовується для документів на яких є дані заповнені людиною на роздрукованих бланках, наприклад під час заповнення медичних анкет.

Після того як всі дані на документах розпізнані а ми маємо доступ до електронного варіанту документів, застосовуються алгоритми екстракції. У

					КНТЕУ 121 06-08.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		14

сучасному програмному забезпеченні коректну екстракцію важливих або потрібних даних можна забезпечити за допомогою розробки проектів, налаштованих спеціально під певний тип документів. Детальніше процес розробки проекту для спеціального типу документів буде розглянуто у розділі 3. Процес екстракції являю собою пошук ключових елементів або слів і використання різних відношень та залежностей для пошуку важливих даних. Дуже простий приклад показано нижче:

ІМ'Я: Максим

Важлива інформація для нас тут це «Максим». Для того щоб знайти її на документі ми спочатку шукаємо ключове слово «ІМ'Я» і налаштовуємо програму таким чином, щоб вона знайшла ключове слово, перемістили вказівник праворуч на один пробіл та екстрагувала те слово, що стоїть наступним. Таким чином, незалежно від того яке значення буде написано після ключового слова, воно буде екстраговане. Ключове слово залишається незмінним на всіх документах.

На виході з даного етапу ми матимемо пари ключ-значення із додатковими параметрами, наприклад: [«Ім'я» : «Максим», { «координати» : «120, 30, 200, 40», «розмір шрифту» : «14» }]. Дані у такому вигляді будуть форматуватися для стадії експорту. Сучасні програмні засоби дозволяють не тільки експорт у більшість популярних форматів (XML, XLS, CSV, SQL, TXT та інші) але й доступ до даних прямо із програмного коду, що дозволяє розробникам окремо обробляти кожен вид даних ще до того, як вони будуть експортовані у вихідний формат. Різні сценарії будуть детальніше розглянуті у наступному підрозділі.

					КНТЕУ 121 06-08.МР	Аркуш
						15
Зм.	Аркуш	№ докум	Підпис	Дата		

1.3. Область використання технологій та огляд існуючих програмних рішень в яких вони використовуються

Існує багато програмних продуктів, що використовують технології OCR та екстракції даних, але величезна кількість з них використовують алгоритми інших. Порівняння програм що використовують одні й ті самі алгоритми було б позбавлено сенсу, тому для початку розглянемо три найвідоміші компанії які показали себе кращими в сфері OCR. Головним критерієм відбору програм стала їх розповсюдженість на ринку програмного забезпечення.

Лідером ринку сьогодні є компанія ABBYY з серією продуктів FineReader. Починаючи з 1989 року програмні продукти компанії широко використовуються бізнесом і користувачами по всьому світу, кількість підтримуваних мов постійно збільшується разом із кількістю користувачів. Є різні продукти для різних цільових аудиторій – користувачів смартфонів, розробників програмного забезпечення, середнього та великого бізнесу. Технології OCR в ABBYY показують найкращі результати при розпізнаванні текстів різними мовами, зокрема української і англійської, що були визначено результатами дослідження.[5]

Для порівняння обрано безкоштовне програмне забезпечення: в першу чергу Tesseract OCR – безкоштовний OCR модуль із великою кількістю реалізацій для різних операційних систем, був розроблений компанією Hewlett Packard у середині 1980-х років. Підтримує багато мов включаючи українську, що є одним із основних критеріїв підбору ПЗ для даного тестування. Модуль розпізнавання Tesseract є найкращим серед безкоштовних рішень і може бути використаний як базис для побудови системи розпізнавання, але для використання його як готового продукту результати розпізнавання не задовольняють стандарти що запропоновані іншими виробниками.

Важливим гравцем на ринку OCR є Google Cloud OCR – розпізнавання тексту від відомого світового виробника ПЗ, має підтримку великої кількості

					КНТЕУ 121 06-08.МР	Аркуш
						16
Зм.	Аркуш	№ докум	Підпис	Дата		

мов, але розповсюджується тільки у вигляді Cloud API, тому даний продукт також можна розглядати як гарний OCR двигун для влаштування в інші продукти.

Оскільки в ході даної роботи розробляється інформаційна система з використанням не тільки OCR технології, а й екстракції слабоструктурованих даних, нижче буде розглянуте рішення, якт вже мають вбудований OCR модулю для подальшої екстракції і обробки даних. Це рішення від вже згаданого раніше розробника ПЗ – ABBYY. Оскільки результати тестувань показали найвищу якість розпізнавання різних типів текстів української мовою, використання продукту від цього виробника є обґрунтованим рішенням для побудови інформаційної системи.

ABBYY FlexiCapture - це високошвидкісна платформа для інтелектуального збору даних та обробки документів, яка може бути успішно використана для екстракції даних з неструктурованих, а також структурованих паперових документів, сканів, електронних повідомлень та інших джерел для подальшого використання в системах управління документами[6]. Чотири основні операції екстракції даних в ABBYY FlexiCapture включають класифікацію, оптичне розпізнавання, перевірку та експорт до систем ERP, ESM, BPM тощо.

ABBYY FlexiCapture здатний задовольнити весь спектр потреб у обробці даних, від невеликих проектів до розподіленого промислового збору великих обсягів даних, забезпечуючи високий рівень безпеки та надійності, що очікується від програмних рішень корпоративного рівня.

Загальний сценарій обробки документів такий:

1. Документи розміщуються в автоматичному пристрої подачі сканера і сканування починається. Зображення також можна імпортувати з електронної пошти, мобільного пристрою тощо.

					КНТЕУ 121 06-08.МР	Аркуш
						17
Зм.	Аркуш	№ докум	Підпис	Дата		

2. Зображення сканованих документів подаються на розпізнавання, після чого програмне забезпечення автоматично ідентифікує документи, включаючи багатосторінкові документи, знаходить значущі поля, проводить екстракцію даних.
3. Після завершення розпізнавання витягнуті дані проходять автоматичну перевірку правил. За необхідності запускається процес візуальної перевірки оператором на наявність даних та помилок правил.
4. Розпізнані та перевірені дані разом з електронними копіями документів зберігаються в базі даних, інформаційній системі або файлі.

Різні документи можна обробляти за один потік. Ви також можете налаштувати програму для обробки документів змішаного типу, наприклад документи, що містять як структуровані, так і слабоструктуровані розділи. Тип документів впливає лише на конфігурацію системи, а саме на метод створення визначень документів. Характер роботи оператора не залежить від типу документів, що обробляються.

У контексті FlexiCapture, слабоструктуровані документи - це документи, що містять набір інформаційних полів, дизайн, кількість та компонування яких можуть суттєво відрізнятися в різних варіантах документа. Ці документи називаються «гнучкими». Наприклад, рахунки - це слабоструктуровані документи, оскільки вони часто різняться як за кількістю позицій, так і за їх форматкуванням через те, що вони надходять від різних компаній. Усі рахунки мають номер та суму, що підлягає оплаті, але розташовані в різних місцях.

Для ідентифікації гнучких форм та вилучення даних із них АБВУУ FlexiCapture використовує гнучкий макет (FlexiLayout). Гнучкий макет створюється за допомогою спеціального модуля під назвою АБВУУ FlexiLayout Studio. Обробка слабоструктурованих документів відрізняється

					КНТЕУ 121 06-08.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

від обробки фіксованих форм лише на етапі створення та завантаження макета у проект.

Адміністратор налаштовує програму для обробки певних типів документів. Щоб підготувати програму для обробки документів, потрібно створити:

1. Проект. Проект - це сукупність налаштувань, пов'язаних з конкретним бізнес-процесом, він включає профілі імпорту зображень, визначення документа та пакети документів для обробки.
2. Одне або декілька визначень документа Створення визначення документа є найважливішим етапом налаштування проекту. Якщо планується обробляти різні типи документів в одному потоці, потрібно створити кілька визначень документів як частину одного проекту.
3. Типи пакетів. Тип пакету визначає, як FlexiCapture буде його обробляти. Типи пакетів включають налаштування, які впливають на те, як зображення розпізнаються та обробляються, як програма перевіряє цілісність пакетів та як вона експортує дані.
4. Профілі імпорту зображень. Профілі імпорту створюються для зручності та впорядкування роботи операторів. Якщо профілі імпорту не налаштовані, кожного разу, коли додається зображення, має бути вибрано джерело імпорту та налаштовано параметри. Профілі імпорту дозволяють додавати зображення одним натисканням кнопки та легко переключатися між джерелами імпорту.
5. Пакет документів. Документи, подані на обробку, додаються до пакетів. Принцип віднесення документів до пакетів визначається користувачем: можливо обробляти всі документи в одному пакеті або формувати пакети з документів, доданих в одну й ту саму дату, або з партії, відсканованої за один прохід. Якщо додати зображення, для

					КНТЕУ 121 06-08.МР	Аркуш
						19
Зм.	Аркуш	№ докум	Підпис	Дата		

якого не визначено тип пакету, налаштування за замовченням будуть використані.

Багато важливих бізнес-процесів залежать від того, що дані швидко передаються відповідним особам або командам, щоб вони могли діяти. Потім інформація, що міститься в цих даних, використовується для прийняття рішення або вводиться у внутрішню систему, щоб рішення або дія могли бути прийняті. Для автоматизації цих бізнес-процесів широко використовуються технології екстракції даних, найбільш поширеними причинами використання цих технологій на підприємствах є:

- Швидша обробка. Вміст із документів, що надходять через будь-який канал, у будь-якому форматі, автоматично витягується, розуміється та доставляється, усуваючи довгий час ручної обробки.
- Огляд існуючих даних та процесів. Інструменти моніторингу та звіти про ключові параметри, включаючи ресурси, продуктивність та точність, забезпечують прозорість та передбачуваність процесів, надаючи адміністраторам розуміння можливостей для вдосконалення.
- Перевірка та контроль нових даних. Критичні поля даних, контекст та сутності визначаються, перевіряються та автоматично обробляються відповідно до бізнес-правил та вимог. Система може бути легко навчена і використовує постійне машинне навчання для постійного вдосконалення та контролю витрат.

Різні організації використовують описані технології з метою автоматизації різних бізнес-процесів, нижче наведено деякі приклади такого використання:

- Банківська організація оцифровує та автоматично класифікує вхідні кредитні заявки з різних каналів. Це дозволяє уникнути втручання

					КНТЕУ 121 06-08.МР	Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата		

людини, оскільки дані обробляються та порівнюються з референтними значеннями автоматично.

- Великі корпорації часто використовують рішення для збору даних, щоб усунути трудомістку роботу, пов'язану з ручною обробкою даних рахунків-фактур та введенням даних про кредит у ERP систему.
- Компанія що спеціалізується на логістиці автоматизувала процес підтвердження доставки для своїх клієнтів за допомогою програмного забезпечення для екстракції слабоструктурованих даних.

1.4. Висновки до розділу 1

Технології розпізнавання символів та OCR є невід'ємною частиною автоматизації документообігу. Їх розвиток починаючи з 1980-х років до сьогодні дозволяє більшому числу організацій і розробників використовувати ці технології у своїх розробках. Сьогодні технології продовжують розвиватися із використанням штучного інтелекту для тренування на різних типах документів і шрифтів. Екстракція слабоструктурованих даних є логічним продовженням розвитку оптичного розпізнавання тексту, після того як паперові документи перейшли у електронний формат і багато компаній вже відмовилися від паперових документів – саме екстракція слабоструктурованих даних дозволяє додати більше автоматизації у багатьох бізнес-процесах, таких як оплата рахунків, ведення бухгалтерії, формування звітів, перевірка заповнених анкет, медичних карток, шкільних тестів тощо. Найкращі результати саме у розвитку даних технологій, не беручи до уваги готові програмні продукти, сьогодні показує компанія ABBYY, і вона надає можливість використовувати свої технології й іншим стороннім виробникам. Після ознайомлення з принципами роботи технологій наступним логічним

					КНТЕУ 121 06-08.МР	Аркуш
						21
Зм.	Аркуш	№ докум	Підпис	Дата		

етапом буде розробка структури бази даних та інтерфейсу користувача, тобто компонентів які напряму не взаємодіють із компонентами розпізнавання та екстракції але є необхідними складовими інформаційної системи.

					<i>КНТЕУ 121 06-08.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		22

РОЗДІЛ 2

ХАРАКТЕРИСТИКА ІНФОРМАЦІЙНОЇ СИСТЕМИ, ПРОЕКТУВАННЯ БАЗИ ДАНИХ ТА ІНТЕРФЕЙСУ КОРИСТУВАЧА

2.1. Опис процедури виконання якої забезпечить інформаційна система

У рамках цієї роботи було створено інформаційну систему для автоматизації процесу обробки лабораторних робіт для студентів і викладачів. Технології оптичного розпізнавання та екстракції даних тут використані з метою валідації роботи на наявність у її тексті необхідних елементів, таких як ім'я студента, тема і номер роботи, група студента

Дана система може бути налаштована для використання із різними типами робіт у різних форматах (офісні документи, зображення, текстові файли) та з різними вимогами. Система складається з чотирьох компонентів:

- Модуль розпізнавання та екстракції даних буде реалізований у вигляді хмарного сервісу на який завантажений заздалегідь розроблений проект у програмі ABBYY FlexiCapture.
- Модуль бази даних MySQL забезпечує зберігання всіх даних інформаційної системи. Серед них є дані користувачів для авторизації, документи із завданнями для лабораторних робіт, виконані роботи, список груп та список дисциплін та інші.
- Інтерфейс користувачів на даному етапі є базовим, розроблений із простою та відомою зв'язкою HTML\CSS що забезпечить доступ до

					<i>КНТЕУ 121 06-08.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			07.09.20	Розробка інформаційної системи оптичного розпізнавання тексту екстракції слабоструктурованих даних	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Гнатченко Д.Д.			07.09.20		<i>P2</i>	<i>23</i>	<i>52</i>
Гарант	Криворучко О.В.			07.09.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив	Дьяченко М.І.			07.09.20				
					<i>Характеристика інформаційної системи, проектування бази даних та інтерфейсу користувача</i>			

інформаційної системи для користувачів без необхідності виконання інсталяції будь-яких додаткових додатків і також буде мати можливість працювати із будь-якого пристрою що має доступ до мережі Інтернет та інстальований веб-браузер.

- Модуль управління системою розроблений на мові PHP останньої версії. Він забезпечує взаємодію всіх компонентів інформаційної системи на сервері.

Система була розроблена і тестована на локальному сервері Apache що входить до складу пакету MAMP для MacOS.

2.2. Етапи проектування бази даних

У якості бази даних використовується СУБД MySQL що входить до пакету MAMP. Нижче надано перелік створених таблиць що забезпечують правильно функціонування інформаційної системи. База даних була розроблена із використанням інтерфейсу PhpMyAdmin що також входить до пакету MAMP.

Першою створювалася таблиця користувачів “users” у якій зберігаються авторизаційні дані. Користувачі системи можуть належати до однієї із трьох ролей:

- Студент
- Староста
- Викладач

Також у логіку системи закладена четверта роль Адміністратора що створює всіх користувачів системи і сам не належить до жодної із ролей.

					КНТЕУ 121 06-08.МР	Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата		

☐	1	id	int(5)	UNSIGNED	Нет	Нет	AUTO_INCREMENT
☐	2	login	varchar(100)	utf8_general_ci	Нет	Нет	
☐	3	name	varchar(50)	utf8_general_ci	Да	NULL	
☐	4	password	varchar(32)	utf8_general_ci	Нет	Нет	
☐	5	reg_date	int(11)	UNSIGNED	Нет	Нет	
☐	6	isHeadman	tinyint(1)		Да	NULL	
☐	7	isTeacher	tinyint(1)		Да	NULL	

Рис. 2.1. Структура полів таблиці "users"

У даній таблиці для розподілу ролей задіяні стовпці "isTeacher" та "isHeadman" для визначення належності користувача до ролей викладача та старости відповідно. Дані поля можуть приймати значення 1 або 0 що відповідає булевим значенням TRUE та FALSE.

Також слід підкреслити що паролі у таблиці зберігаються із використанням функції хешування MD5 для забезпечення додаткової безпеки, у відкритому вигляді паролі не зберігаються та не передаються.

Кожен користувач системи має свій унікальний ідентифікатор для простоти позначення цього користувача у інших таблицях бази даних.

id	login	name	password	reg_date	isHeadman	isTeacher
7	starosta@email.com	Староста	202cb962ac59075b964b07152d234b70	1606459990	1	NULL
8	vykladach@email.com	Викладач	4297f44b13955235245b2497399d7a93	1606465309	NULL	1
9	student@email.com	Студент	4297f44b13955235245b2497399d7a93	1606465540	NULL	NULL

Рис. 2.2. Таблиця "users" заповнена трьома користувачами що належать до різних ролей.

Таблиця для зберігання груп "groups" є простішою зо своєю структурою і метою, вона зберігає інформацію про назву навчальних груп та відповідність до старост.

					КНТЕУ 121 06-08.МР	Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата		

Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
id 	int(11)			Нет	Нет		AUTO_INCREMENT
headManId	int(5)			Да	NULL		
groupName	varchar(50)	utf8_general_ci		Нет	Нет		

Рис. 2.3. Структура таблиці “groups”

id	headManId	groupName
1	7	6м

Рис. 2.4. Дані у таблиці “groups”

Як видно на рисунку 2.4, поле “headManId” має значення 7, що відповідає ідентифікатору старости на рисунку 2.2. Таким чином система встановлює відповідність між старостами та групами.

У таблиці “subjects” зберігається інформація про наявні у системі дисципліни. Дисципліни у системи додаються користувачами із правами викладачів для того, щоб кожне завдання і кожна робота були прив’язані до конкретної навчальної дисципліни.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 	int(5)			Нет	Нет		AUTO_INCREMENT
2	name	varchar(255)	utf8_general_ci		Нет	Нет		
3	teacherId	int(255)			Да	NULL		

Рис. 2.5. Структура таблиці “subjects”

Ця таблиця має три поля, що відповідають унікальному ідентифікатору дисципліни, її назві та унікальному ідентифікатору викладача, який створив дану дисципліну у системі. Таблиця має зв’язок із таблицею “users” для визначення ідентифікатора викладача.

					КНТЕУ 121 06-08.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		26

id	name	teacherId
1	Безпека Інтернет-ресурсів	8

Рис. 2.6. Дані у таблиці “subjects”

Значення поля “teacherId” на рисунку 2.6 відповідає полю id с рисунку 2.2 таблиці “users”.

Таблиця, що містить інформацію про доступні завдання має назву “tasks” та на відміну від раніше описаних таблиць має не тільки дані, що описують користувачів та завдання, але також і містять шлях до розміщення файлу завдання на сервері, де розміщена інформаційна система.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 	int(5)			Нет	Нет		AUTO_INCREMENT
2	subjectId	varchar(5)	utf8_general_ci		Да	NULL		
3	taskNumber	varchar(5)	utf8_general_ci		Нет	Нет		
4	filePath	varchar(255)	utf8_general_ci		Нет	Нет		

Рис. 2.7. Структура таблиці “tasks”

Таблиця має чотири поля у своїй структурі. Поле “id” зберігає унікальний ідентифікатор дисципліни, поле “subjectId” використовується для прив’язки завдання до конкретної дисципліни (з рисунку 2.6), поле “taskNumber” відповідає номеру лабораторної роботи, а останнє поле “filePath” зберігає шлях до файлу із завданням на сервері. Цей файл викладач завантажив на сервер за допомогою свого інтерфейсу.

					КНТЕУ 121 06-08.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		27

id	subjectId	taskNumber	filePath
3	1	1	./uploaded_files/tasksc90b1680677bf79c68ad0c75c9f6...
4	1	2	./uploaded_files/tasks780a7e66ce007d0432034ceb91e7...

Рис. 2.8. Дані таблиці “tasks”

Як можна побачити на рисунку 2.8, на даний момент у системі розміщено два завдання для лабораторних робіт, що відносяться до однієї дисципліни.

Останньою та найбільш складною у розрізі зв'язку з іншими таблицями є таблиця лабораторних робіт - “labs”. Таблиця має 6 полів. Поле “id” є унікальним ідентифікатором виконаної роботи, поле “student” містить ім'я студента, поле “topic” зберігає інформацію із темою лабораторної роботи, “labNumber” містить номер роботи у дисципліні, поле “grade” міститиме оцінку виставлену викладачем студенту за виконання цієї роботи, а поле “filePath” містить шлях до роботи на сервері, що у подальшому використовується як посилання на завантаження цієї роботи викладачем для перевірки.

Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
id 	int(5)			Нет	Нет		AUTO_INCREMENT
student	varchar(100)	utf8_general_ci		Нет	Нет		
topic	varchar(255)	utf8_general_ci		Нет	Нет		
labNumber	varchar(255)	utf8_general_ci		Нет	Нет		
grade	int(3)			Да	NULL		
filePath	varchar(255)	utf8_general_ci		Нет	Нет		

Рис. 2.9. Структура таблиці “labs”

					КНТЕУ 121 06-08.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		28

id	student	topic	labNumber	grade	filePath
1	Дьяченко Максим	Налаштування міжмережевого екрану	1	NULL	./uploaded_files/5345b163a6c74

Рис. 2.10. Дані у таблиці “labs”

Особливу увагу слід звернути на те, що поля “student”, “topic” та “labNumber” не вводяться користувачем, а зчитуються автоматично з роботи за допомогою використання модулю розпізнавання та екстракції даних. Саме це є головною ідеєю інформаційної системи – продемонструвати можливість автоматизації процесу ручного вводу даних у систему. Слід також зазначити, що кількість полів для розпізнавання і кількість відповідних стовпців може бути розширено для забезпечення відповідності вимогам конкретної дисципліни або конкретного університету або іншого навчального закладу.

Таким чином ми маємо СУБД що містить одну базу даних, у складі якої міститься п’ять таблиць необхідних для зберігання і обробки даних у системі.

2.3. Етапи проектування інтерфейсу користувача

Для проектування інтерфейсу користувача було використано програмне забезпечення Figma – графічний редактор, що поставив себе як одне із кращих рішень на ринку, що безкоштовне надає можливість дизайнерам та розробникам моделювати інтерфейс для своїх систем.

Кінцевий проект має у своєму складі 12 видів доступних для різних користувачів та адміністратора. Інтерфейс є простим та у ньому наявні лише необхідні елементи для виконання всіх запланованих функцій інформаційної системи.

					КНТЕУ 121 06-08.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		29

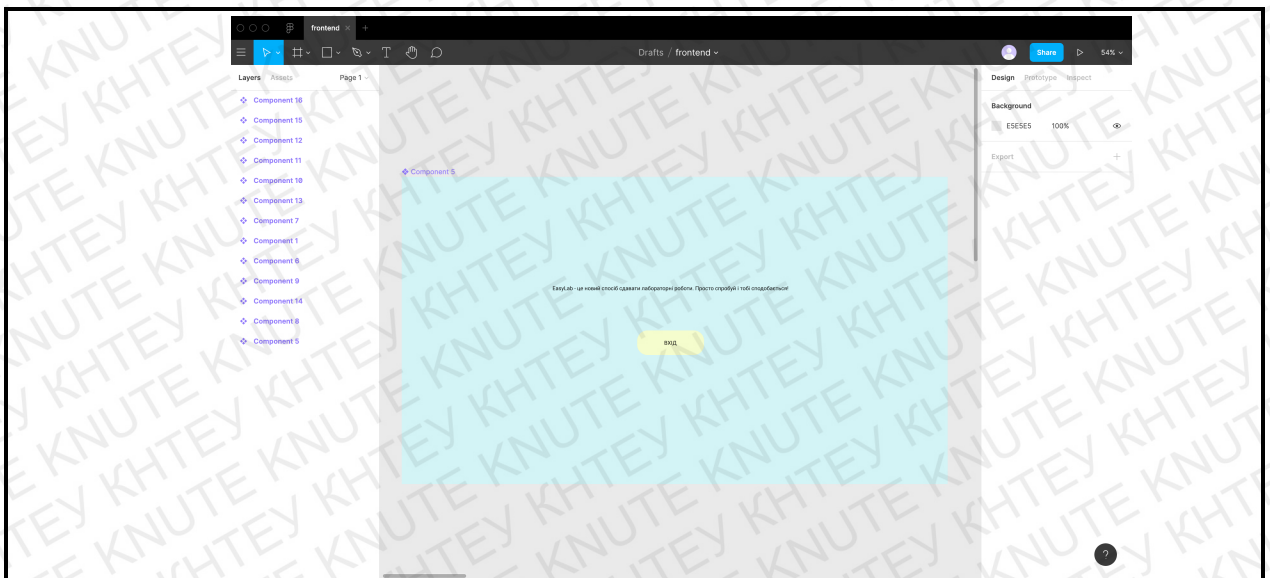


Рис. 2.11. Головна сторінка

На головній сторінці розміщений привітальний текст та посилання до сторінки авторизації.

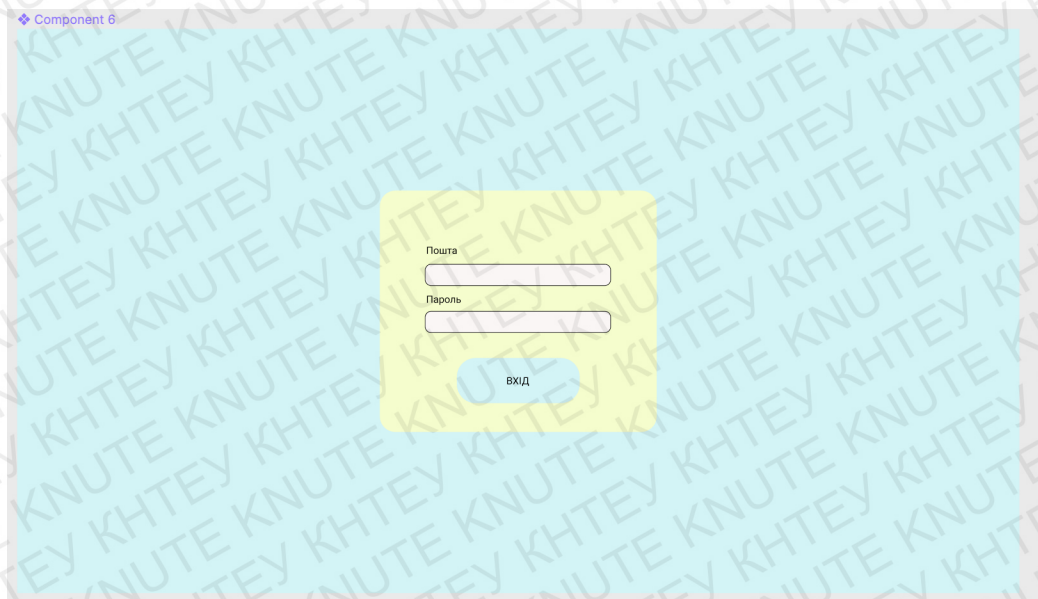


Рис. 2.12. Сторінка авторизації

Сторінка авторизації містить два поля: для вводу пошти, як унікального ідентифікатора користувача, та поле для вводу пароля.

					<i>КНТЕУ 121 06-08.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		30

Далі, в залежності від того, до якої ролі належить користувач, він перенаправляється до відповідної сторінки інтерфейсу.

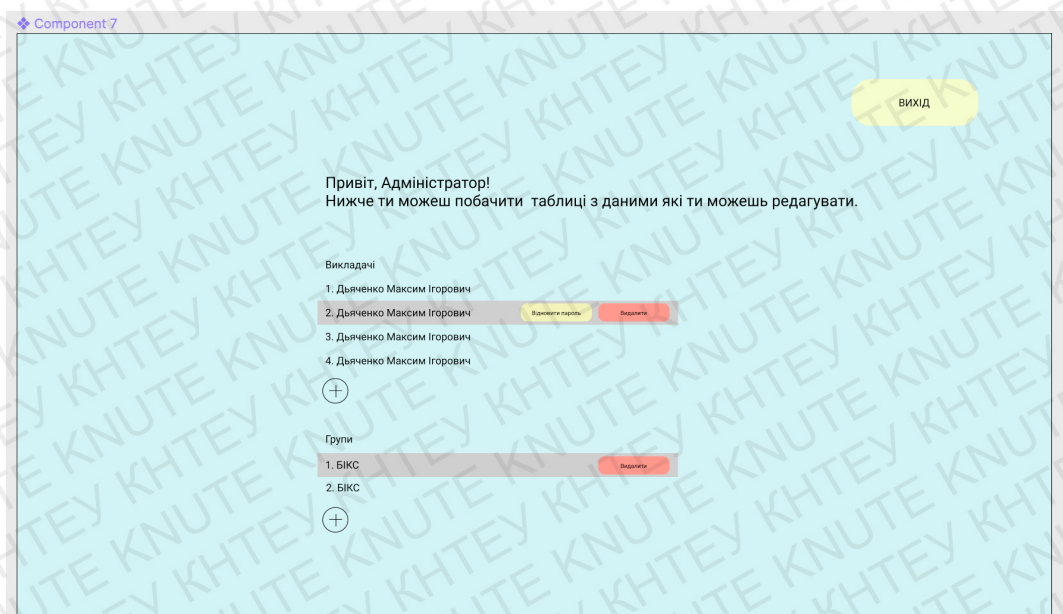


Рис. 2.13. Інтерфейс адміністратора

Адміністратор системи має повноваження додавати до системи викладачів та групи. Роль адміністратора, за логічним дизайном системи, має можливість додавати будь-які дані до будь-яких таблиць за необхідності, але перша версія продукту має вбудовані лише необхідні для базового функціонування елементи.

Component 8

вихід

Ім'я

Пошта

Додати викладача

Рис. 2.14. Додавання викладача до системи.

Component 9

вихід

Назва

Пошта старости

Додати групу

Рис. 2.15. Додавання групи до системи.

Разом із створенням групи до системи додається її староста. Паролі генеруються системою автоматично та відправляються на електронну пошту користувача.

					КНТЕУ 121 06-08.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		32

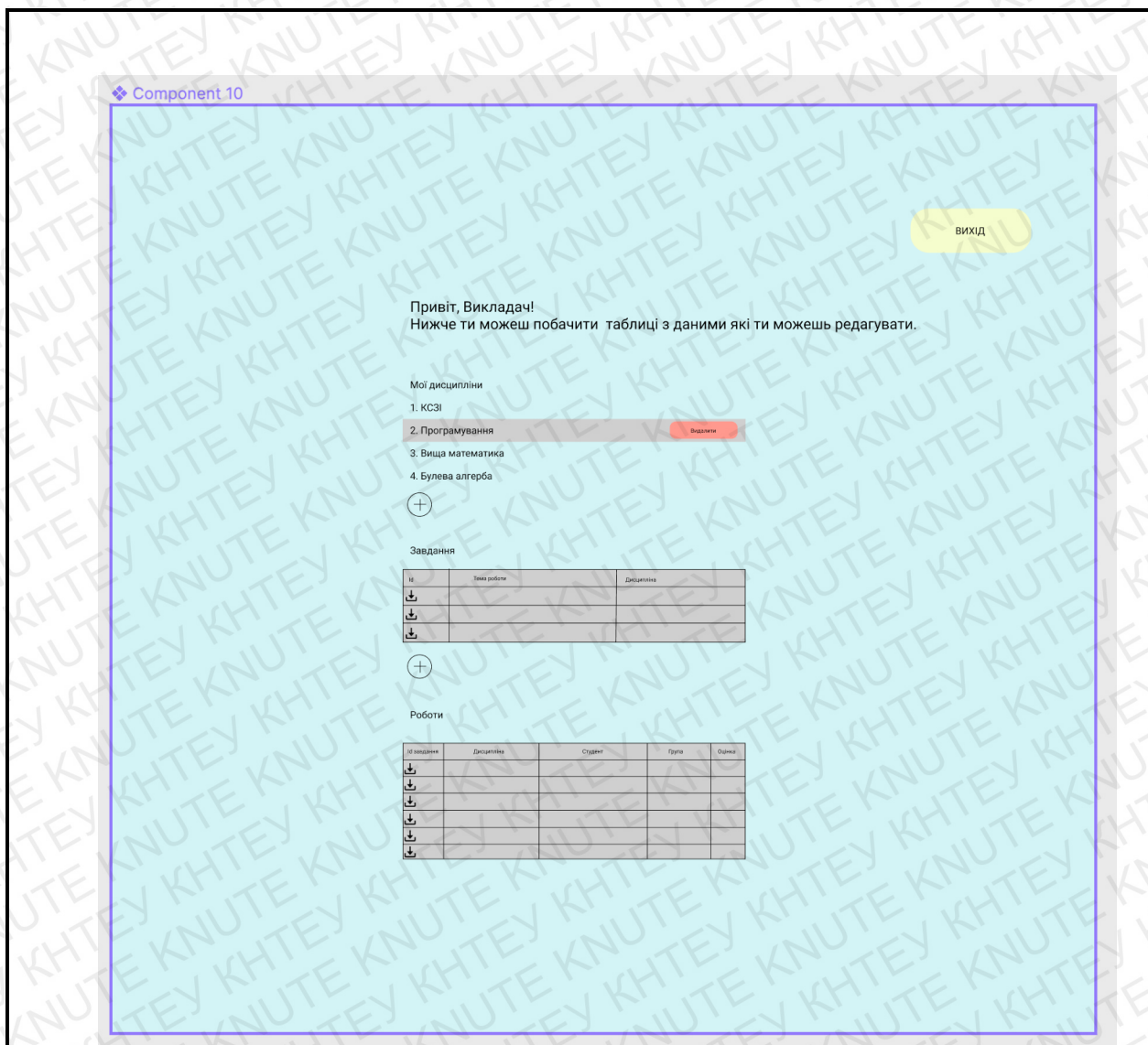


Рис. 2.16. Інтерфейс викладача

Викладач має можливість створювати дисципліни, завдання, а також бачити список зданих робіт і завантажувати їх для перевірки. Після перевірки роботи викладач виставляє оцінку.

Component 11

Вихід

Назва

Групи

Додати дисципліну

Рис. 2.17. Додавання дисциплін

Component 12

Вихід

Тема

Дисципліна

Вибрати локальний файл або перетягніть його в цю зону

Додати групу

Рис. 2.18. Додавання завдань

Викладач має можливість вибрати групу та дисципліну із наявних у базі даних.

					КНТЕУ 121 06-08.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		34

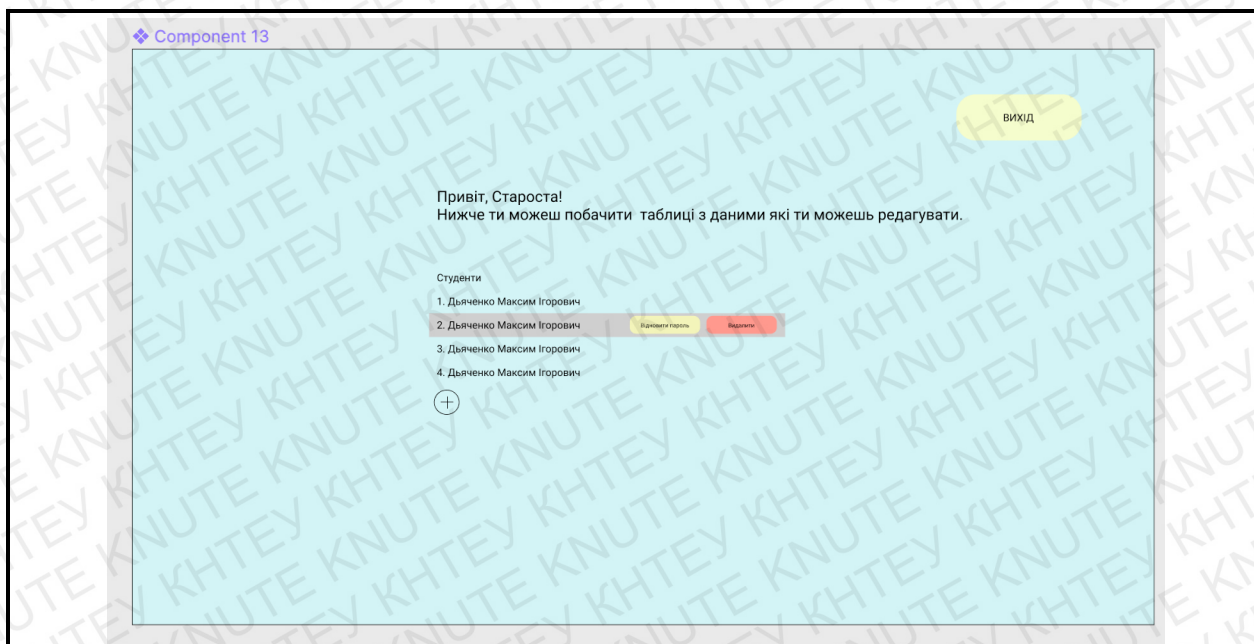


Рис. 2.19. Інтерфейс старости

Роль старости створена у системі для додавання студентів. Аналогічно до додавання викладачів адміністратором, після введення пошти користувача його пароль генерується автоматично та надсилається на електронну пошту.

Рис. 2.20. Сторінка додавання студентів

Останнім необхідним елементом інтерфейсу системи є інтерфейс студента. Студент має можливість переглядати список своїх дисциплін та доступних завдань до лабораторних робіт, а також переглядати список своїх зданих робіт і виставлених оцінок.

Найважливішою з точки зору тематики даної роботи є сторінка здачі лабораторної роботи, адже вона зв'язує інтерфейс користувача, хмарний сервіс розпізнавання та екстракції даних та базу даних.

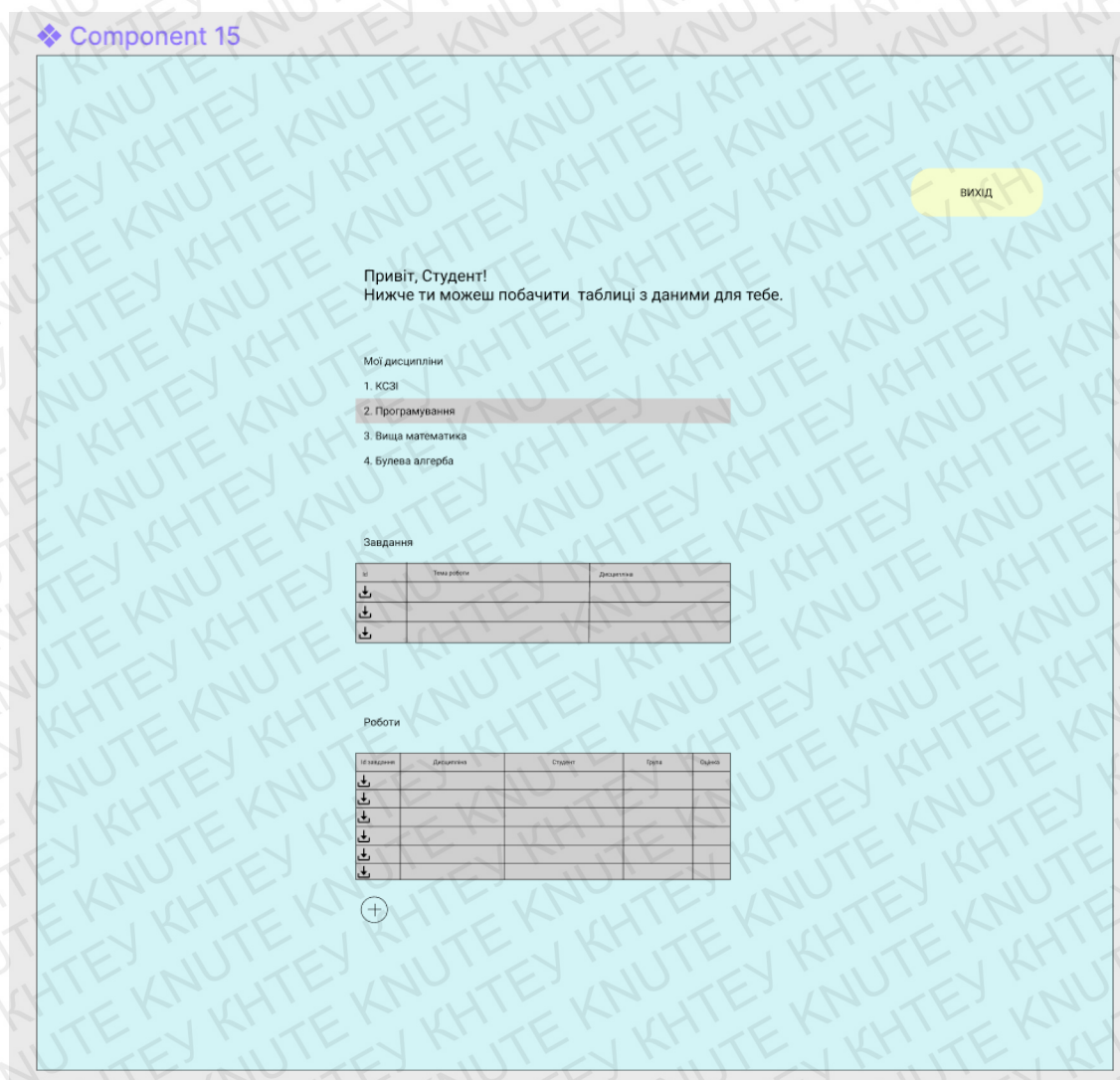


Рис. 2.21. Інтерфейс студента

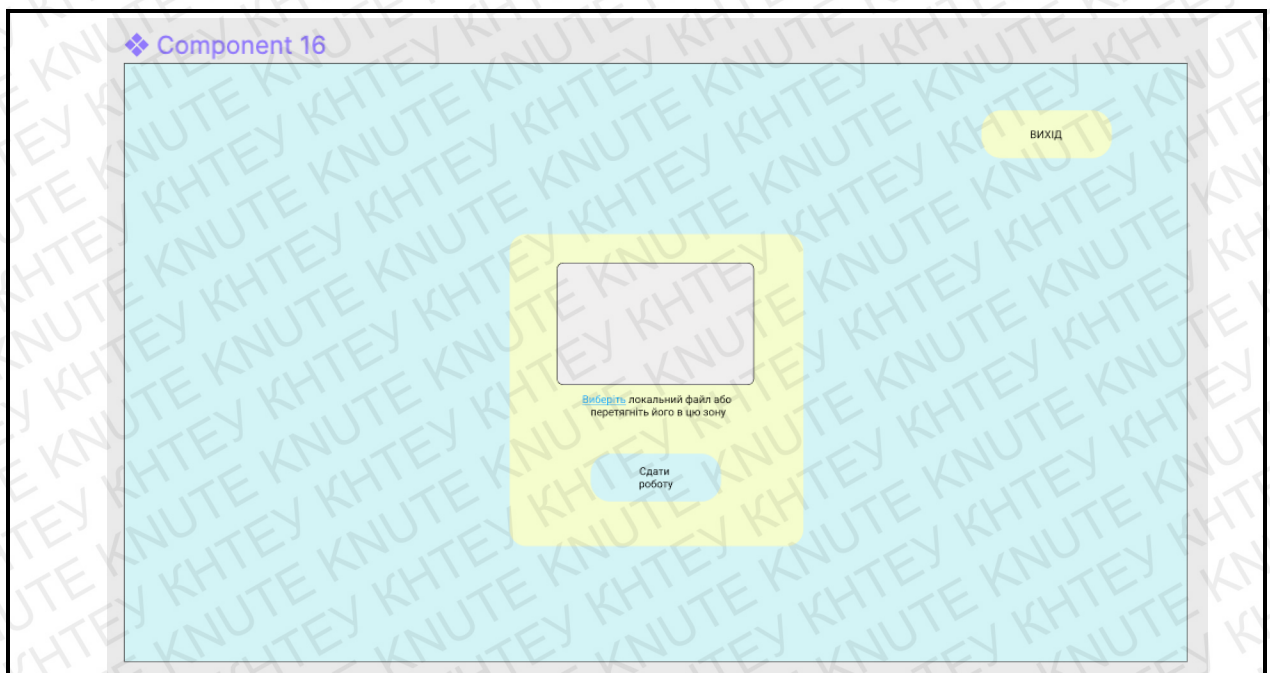


Рис. 2.22. Сторінка здачі лабораторної роботи.

2.4. Висновки до розділу 2

У другому розділі були розглянуті два компоненти інформаційної системи, які можна назвати другорядними до тематики даної роботи, але є необхідними для її коректного функціонування. Тепер, маючи базу даних та інтерфейс користувачів, описаних у цьому розділі, а також ознайомившись із технологіями розпізнавання та екстракції даних у першому розділі, можна переходити до основної частини розробки програмного продукту, що об'єднає всі компоненти та забезпечить їх коректну взаємодію.

					КНТЕУ 121 06-08.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		37

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО МОДУЛЮ УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ СИСТЕМОЮ

3.1. Опис програмного модулю з управління інформаційною системою та коду її інтерфейсу

Програмний модуль управління був написаний на мові програмування PHP без використання додаткових готових модулів. Цей модуль складається з:

- Систем виконання API запитів на віддалений сервер
- Систем запитів до бази даних для читання і запису інформації
- Додаткових модулів для взаємодії із інтерфейсом користувача

Тобто, три підсистеми для взаємодії з компонентами загальної системи.

Для виконання API запитів до хмарного сервісу, де виконується обробка, розпізнавання та екстракція даних з документів, було використано вбудований у PHP модуль CURL. Для пришвидшення процесу розробки та тестування, всі запити були змодельовані у додатку PostMan і автоматично переведені у вигляд, готовий для сприймання у мові PHP.

FlexiCapture Cloud, як система розпізнавання та екстракції даних, надає широкі можливості для інтеграції із іншими системами, і в рамках даної розробки було досліджено можливості їх програмного інтерфейсу взаємодії API. Весь процес складається з восьми запитів до віддаленого серверу:

1. Створити сесію
2. Створити новий пакет документів

					<i>КНТЕУ 121 06-08.МР</i>					
Зм.	Аркуш	№ докум.	Підпис	Дата						
Зав. каф.	Криворучко О.В.			19.10.20	Розробка інформаційної системи оптичного розпізнавання тексту й екстракції слабоструктурованих даних	Стадія	Аркуш	Аркушів		
Керівник	Гнатченко Д.Д.			19.10.20		РЗ	38	52		
Гарант	Криворучко О.В.			19.10.20		Факультет інформаційних технологій 2м курс, 6 група				
Розробив	Дьяченко М.І.			19.10.20						

3. Відкрити пакет
4. Додати документ до пакету
5. Закрити пакет
6. Обробити документ
7. Отримати посилання на документ з результатами
8. Завантажити документ з результатами

Всі запити представляють собою HTTP POST запити у JSON форматі. Вихідні дані надсилаються на сервер також у форматі JSON. Документи для відправлення кодуються функцією Base64 для представлення у текстовому вигляді.

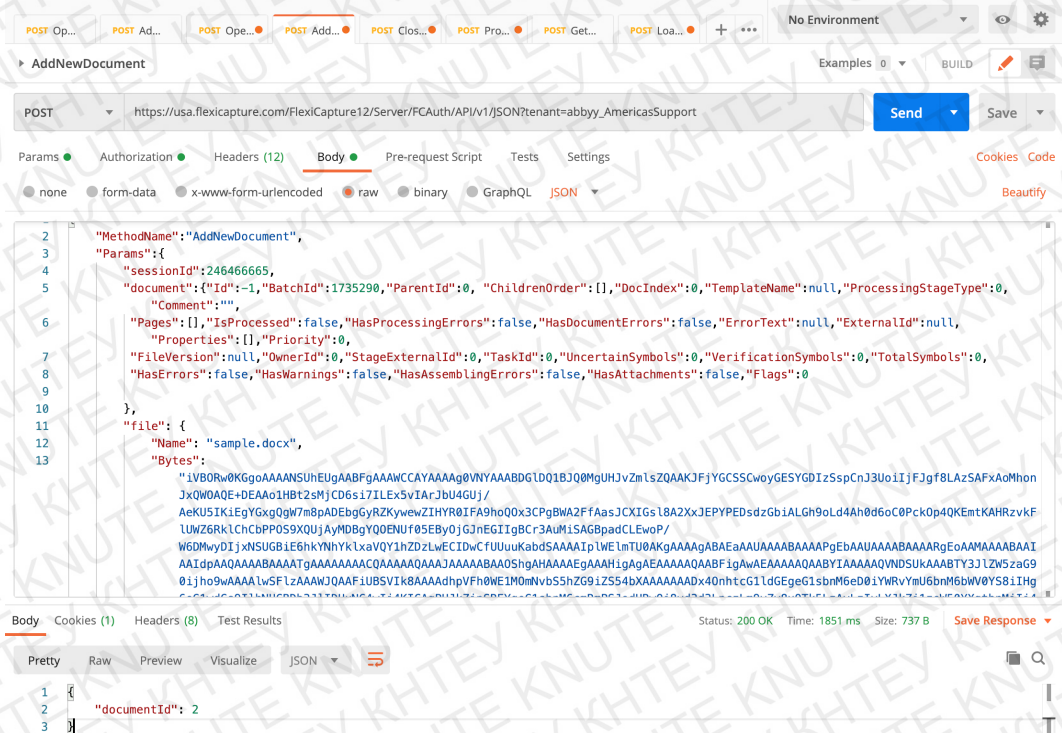


Рис. 3.1. POST запит для додавання документу до пакету обробки

						Аркуш
						39
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06-08.МР	

Після того, як всі запити розроблені і протестовані, вони перекладаються у вигляд PHP коду та вставляються у вигляді функцій до основного коду програми.

Далі ці функції викликаються одна за одною і результати виконання однієї передаються до наступної, як цього вимагає документація модулю.

Розробивши систему API запитів і отримавши результати обробки, можна переходити до розробки системи запитів до бази даних для запису та читання даних. У системі використана база даних MySQL, розроблена у другому розділі даної роботи.

```
function openSession () {  
  
function addNewBatch ($sessionId) {  
  
function openBatch ($sessionId, $batchId) {  
}  
  
function addNewDocument ($sessionId, $batchId, $documentBytes) {  
}  
  
function closeBatch ($sessionId, $batchId) {  
}  
  
function processBatch ($sessionId, $batchId) {  
}  
  
function getResultList ($sessionId, $batchId, $documentId) {  
  
    $curl = curl_init();  
  
    curl_setopt_array($curl, array(  
        CURLOPT_URL => "https://usa.flexicapture.com/FlexiCapture12/Server/FCAuth/API/v1/JSON?tenant=abbyy_AmericasSupport",  
        CURLOPT_RETURNTRANSFER => true,  
        CURLOPT_ENCODING => "",  
        CURLOPT_MAXREDIRS => 10,  
        CURLOPT_TIMEOUT => 0,  
        CURLOPT_FOLLOWLOCATION => true,  
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,  
        CURLOPT_CUSTOMREQUEST => "POST",  
        CURLOPT_POSTFIELDS => "{\n    \"MethodName\": \"GetDocumentResultsList\",\n    \"Params\": {\n        \"sessionId\": \"$sessionId\", \n        \"batchId\": \"$batchId\", \n        \"documentId\": \"$documentId\" \n    }\n    }",  
        CURLOPT_HTTPHEADER => array(  
            "Authorization: Basic YXBpX3VzZXI6cGZc3dvcnQx",  
            "Content-Type: application/json",  
            "Cookie: FlexiCaptureTmpPrn=Ticket=CDf283C6EE9221C65D7DED31E8DFDB7461C99FEB1D3D8C7491B9288D7BECCF3D8A83A6BD189924F888ECA36AC9951B019A99C81BDDDB5465DA502EA017B4942D099E285F0CBA865B8CD09D56D239FAFF",  
        ),  
    );  
  
    $response = curl_exec($curl);  
  
    curl_close($curl);  
    $arr = json_decode($response, true);  
    return $arr;  
}  
  
function loadDocumentResult ($sessionId, $batchId, $fileName, $documentId) {
```

Рис. 3.2. Модуль HTTP POST запитів до віддаленого серверу

					КНТЕУ 121 06-08.МР	Аркуш
						40
Зм.	Аркуш	№ докум	Підпис	Дата		

```

$sessionId = openSession();
$batchId = addNewBatch($sessionId);
// //echo $batchId;
openBatch($sessionId, $batchId);
$file = file_get_contents($dest_path);
$documentBytes = base64_encode($file);
$documentId = addNewDocument($sessionId, $batchId, $documentBytes);
closeBatch($sessionId, $batchId);
processBatch($sessionId, $batchId);
echo 'session: ' . $sessionId . ' document: ' . $documentId . 'batch: ' . $batchId;
sleep(30);
$fileName = getResultsList($sessionId, $batchId, $documentId);
print_r($fileName);
$fileNameToDownload = $fileName['fileNames'][0];
echo $fileNameToDownload;

$fileNameToDownload = str_replace('\\', '\\\\\\\\', $fileNameToDownload);
echo $fileNameToDownload;

$resultBytesArray = loadDocumentResult($sessionId, $batchId, "Export\\\\TESTbatch\\\\Data_00000002.json", 2);
print_r($resultBytesArray);
$resultBytes = $resultBytesArray['file']['Bytes'];
$decodedJson = base64_decode($resultBytes);
file_put_contents('Export\\TESTbatch\\Data_00000002.json', $decodedJson);

```

Рис. 3.3. Логіка виконання запитів та запису результатів до файлу.

Запити до бази даних виконуються функцією `mysqli()`, що також є вбудованою у мову PHP. Нижче наведено приклад, що читає отриманий результат у форматі JSON та записує їх у відповідну таблицю “labs”.

```

$string = file_get_contents('Export\\TESTbatch\\Data_00000002.json');
$json_a = json_decode($string, true);

print_r($json_a);

echo "<br><br>";

$labNumber = $json_a['Documents'][0]['DocumentData']['Fields'][0]['Value'];
$studentName = $json_a['Documents'][0]['DocumentData']['Fields'][1]['Value'];
$topic = $json_a['Documents'][0]['DocumentData']['Fields'][2]['Value'];

function addLabToDb($student, $topic, $labNumber, $filePath) {
    DEFINE('DB_USERNAME', 'root');
    DEFINE('DB_PASSWORD', 'root');
    DEFINE('DB_HOST', 'localhost');
    DEFINE('DB_DATABASE', 'mybase');

    $mysqli = new mysqli(DB_HOST, DB_USERNAME, DB_PASSWORD, DB_DATABASE);

    if (mysqli_connect_error()) {
        die('Connect Error ('.mysqli_connect_errno().') '.mysqli_connect_error());
    }

    $mysqli->query("INSERT INTO `labs` (`student`, `topic`, `labNumber`, `filePath`) VALUES ('" . $student . "', '" . $topic . "', '" . $labNumber . "', '" . $filePath . "')");

    $mysqli->close();
}

addLabToDb($studentName, $topic, $labNumber, $dest_path);

```

Рис. 3.4. Читання та запис результатів у базу даних

Читання та запис у базу даних використовуються для виконання всіх основних функцій програми, таких як:

					КНТЕУ 121 06-08.МР	Аркуш
						41
Зм.	Аркуш	№ докум	Підпис	Дата		

- Створення нового користувача
- Створення дисципліни
- Створення завдання
- Читання списку задач
- Читання списку виконаних робіт
- Читання списку груп, студентів, викладачів та інше.

Наступним етапом буде розробка додаткових модулів, таких як модулі авторизації користувачів, перевірки та хешування паролів. Для перевірки відповідності користувача до певної ролі написані функції `isTeacher()` та `isHeadman()`, о наведені на рисунку 3.5. Повний код програми наведений у додатку А.

```
function isHeadman ($result_set) {
    while(($row = $result_set->fetch_assoc()) != false){
        return $row['isHeadman'];
    }
}

function isTeacher ($result_set) {
    while(($row = $result_set->fetch_assoc()) != false){
        return $row['isTeacher'];
    }
}
```

Рис. 3.5. Функції `isTeacher()` та `isHeadman()`.

Наведені функції читають відповідні колонки бази даних “users” і у разі появи там значення 1 повертають булеве `true`, що далі перенаправляє користувача на відповідну сторінку програми. На рисунку 3.6 зображено код, що здійснює перенаправлення авторизованих користувачів на їх сторінки.

					КНТЕУ 121 06-08.МР	Аркуш
						42
Зм.	Аркуш	№ докум	Підпис	Дата		

```

$user = getUser();
$login = getLogin($user);

if ($login == null) {
    header("Location: user_not_exist.php");
}

if ($user != null){
    $user = getUser();
    $isHeadman = isHeadman($user);
    if ($isHeadman == 1) {
        header("Location: headman.php");
    }
    $user = getUser();
    $isTeacher = isTeacher($user);
    if ($isTeacher == 1) {
        header("Location: teacher.php");
    } else {
        header("Location: student.php");
    }
}

```

Рис. 3.6. Модуль перенаправлення користувачів

Після того, як розроблена основна логіка системи, забезпечена її взаємодія із віддаленими сервісами та базою даних, останньою задачею є розробка інтерфейсу користувачів за макетом, описаним у другому розділі. Для цього була використана мова розмітки гіпертексту HTML із вставками програмного коду на PHP. Всі сторінки системи мають схожу структуру, це ввід-вивід даних та взаємодія із сервером. Для прикладу нижче наведено програмний код сторінки адміністратора із виведенням таблиці користувачів системи (Рис. 3.7).

					<i>КНТЕУ 121 06-08.МР</i>	Аркуш
						43
Зм.	Аркуш	№ докум	Підпис	Дата		

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="main_style.css">
  <link rel="stylesheet" href="normalize.css">
  <title>EasyLab</title>
</head>
<body>
  <section class="main">
    <a href="signin.php" class="sign_out"><button class="sign_out" style="background-color: #F4FDC0;">Вихід</button></a>
    <p>Привіт, Адміністратор!
    Нижче ти можеш побачити таблиці з даними які ти можеш редагувати.</p>

<?php
function getResult ($result_set) {
    while(($row = $result_set->fetch_assoc()) != false){
        echo $row[id], ' ', $row[login];
        echo "<br>";
    }
}

function getUsers () {
}

getUsers();
?>

<table class="groups">
  <tr>
    <td class="number">1.</td>
    <td class="gr_name">БІК</td>
    <td><button class="table_button delete">Видалити</button></td>
  </tr>
  <tr>
    <td class="ad"><a href="add_group.html"><button class="add">+</button></a>
  </tr>
</table>

</section>
</body>
</html>

```

Рис. 3.8. Код інтерфейсу адміністратора

За допомогою функції getUsers() система робить запит да бази даних і отримую список наявних користувачів, а за допомогою функції getResult() ці дані відображаються у інтерфейсі.

3.2. Опис модулю розпізнавання тексту та екстракції даних

Модуль розпізнавання тексту та екстракції даних розроблений з використання програмного забезпечення ABBYY FlexiCapture Cloud. Це платне ПЗ, але для цієї роботи компанією ABBYY було надано ліцензію для тимчасового використання.

Створення проекту в ABBYY FlexiCapture складається з таких етапів, у лапках написані терміни притаманні саме ПЗ ABBYY:

- Розробка «гнучкого опису» документу у FlexiLayout Studio
- Створення нового проекту і розробка «визначення документа» на основі створеного «гнучкого опису»
- Локальне тестування проекту

					КНТЕУ 121 06-08.МР	Аркуш
						44
Зм.	Аркуш	№ докум	Підпис	Дата		

- Завантаження проекту у хмару ABBYY

Після того як проект завантажено у хмару їм можна управляти за допомогою API, що було описано у розділі 2.

Для створення «гнучкого опису» було необхідно відкрити додаток FlexiLayout Studio та взяти декілька прикладів документів, які ми збираємося розпізнавати та вилучати дані. Для цього були використані реальні лабораторні роботи студентів. Спочатку, для ідентифікації того, що елемент належить саме до типу «звіт з лабораторної роботи» створюється елемент, що буди шукати слово звіт. У інтерфейсі програми це виглядає так, як показано на рисунку 3.9. Після цього можна починати пошук поступних елементів, а саме тих що ми збираємося екстрагувати із документи, у даному випадку це буде номер роботи, тема роботи та ім'я студента.

Для пошуку номер роботу спочатку ми знайдемо слова «лабораторна робота» або «лабораторної роботи», а потім налаштуємо проект таким чином, щоб у наступному полі виконувався пошук цифри, яка стоїть праворуч від тих слів. Таким чином пошук не буде прив'язаний до конкретних координат, але перевірятиметься наявність елемента на сторінці. Це продемонстровано на рисунках 3.10 та 3.11.

					КНТЕУ 121 06-08.МР	Аркуш
						45
Зм.	Аркуш	№ докум	Підпис	Дата		

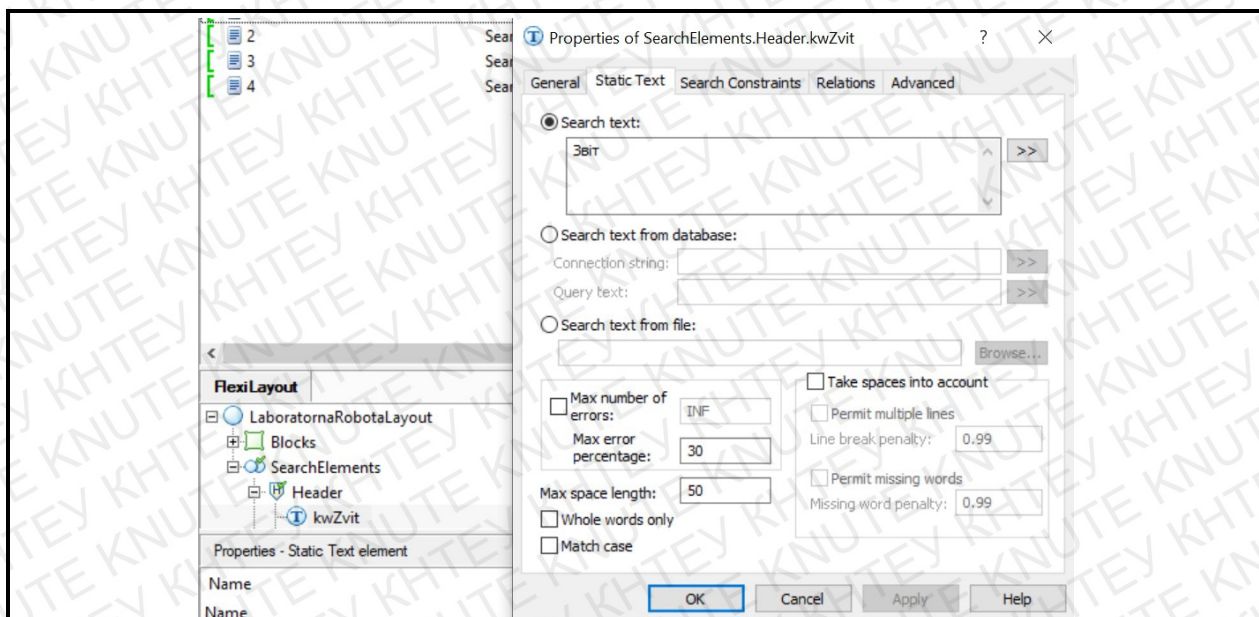


Рис. 3.9. Створення елементу, що шукає слово «звіт»

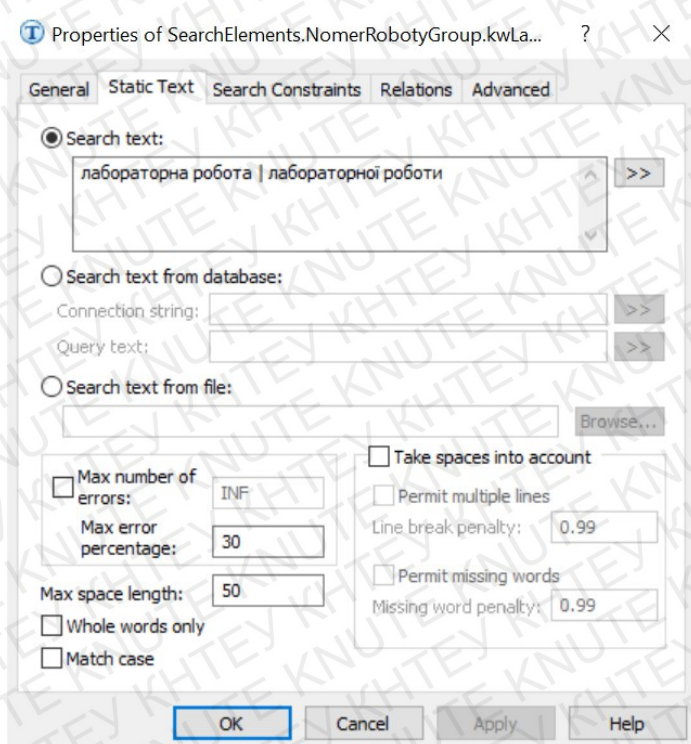


Рис. 3.10. Пошук ключових слів «лабораторна робота»

							Аркуш
						<i>КНТЕУ 121 06-08.МР</i>	46
Зм.	Аркуш	№ докум	Підпис	Дата			

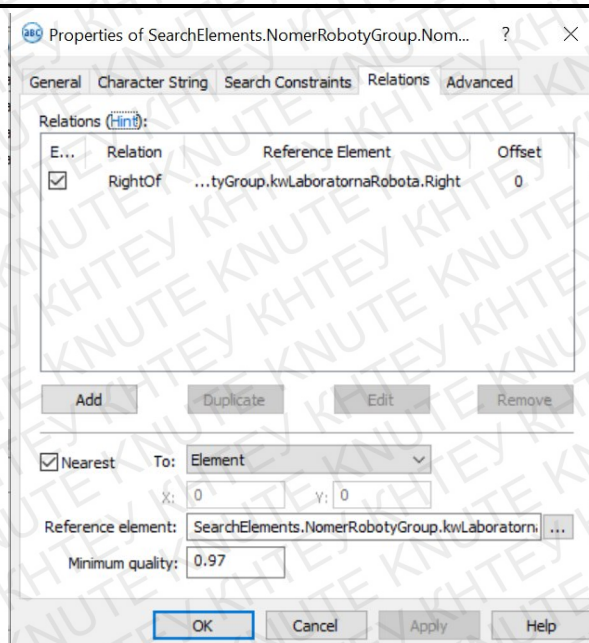


Рис. 3.11. Пошук номера лабораторної роботи.

Після цього аналогічним чином ми знаходимо елементи, що відповідатимуть імені студента та темі лабораторної роботи. Результати накладання «гнучкого опису» на документи з різною структурою демонструються на рисунках 3.12 та 3.13

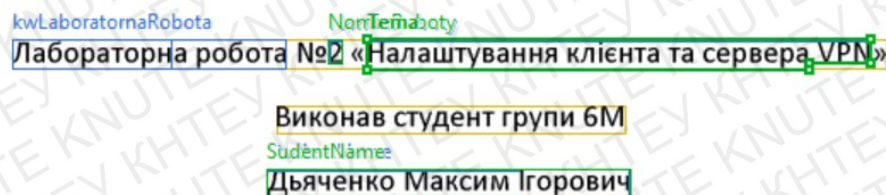


Рис. 3.12. Накладання проекту на документ(а)

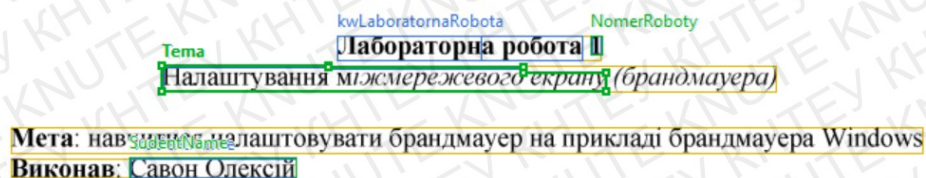


Рис. 3.13. Накладання проекту на документ(б)

					КНТЕУ 121 06-08.МР	Аркуш
						47
Зм.	Аркуш	№ докум	Підпис	Дата		

Після цього проект FlexiLayout експортується та створюється проект FlexiCapture. У інтерфейсі користувача ми можемо побачити список полів, які будуть шукатися на документах.

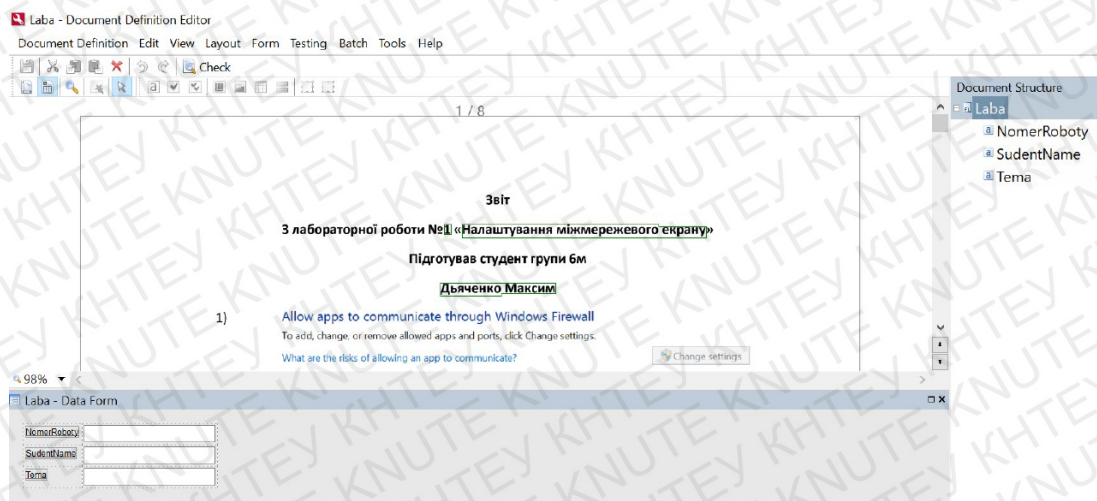


Рис. 3.14. «Визначення документа» у проекті FlexiCapture

Наступним етапом буде завантаження проекту на віддалений сервер АВВУУ, яке теж виконується за допомогою інтерфейсу FlexiCapture. Після цього проект готовий до взаємодії із написаним раніше кодом .

3.3. Висновки до розділу 3

Було розроблений програмний код управління системою на мові програмування PHP, створений інтерфейс користувача на HTML та розроблений FlexiCapture проект, який взаємодіє з іншими компонентами за допомогою інтерфейсу API. Інформаційна система готова до використання.

					КНТЕУ 121 06-08.МР	Аркуш
						48
Зм.	Аркуш	№ докум	Підпис	Дата		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

З проведених досліджень можна зробити наступні висновки:

1. Проведено дослідження технологій оптичного розпізнавання тексту та екстракції слабоструктурованих даних.
2. Розроблена інформаційна система може бути модифікована для використання у будь-якому закладі освіти для автоматизації процесу здачі та валідації лабораторних робіт студентів.
3. Створення веб-додатку дозволяє зробити його доступним для будь-якого пристрою на будь-якій операційній системі без необхідності інсталяції додаткового ПЗ для користувачів.
4. В процесі виконання магістерської роботи розроблено інформаційну систему для автоматизації процесів.
5. Розроблена система повністю задовольняє поставлені функціональні вимоги.
6. У подальших дослідженнях можна додати такі функції як перевірка робіт на антиплагіат, порівняння зданих робіт для виключення можливості копіювання, додавання більшої кількості полів для валідації та глибшого аналізу роботи без людського втручання, додати можливість налаштування кінцевих дат здачі для робіт.

					<i>КНТЕУ 121 06-08.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		30.10.20	Розробка інформаційної системи оптичного розпізнавання тексту й екстракції слабоструктурованих даних	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Гнатченко Д.Д.		30.10.20		<i>ВП</i>	<i>49</i>	<i>52</i>
Гарант		Криворучко О.В.		30.10.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив		Дьяченко М.І.		30.10.20				
					<i>Висновки та пропозиції</i>			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 1.Оптичне розпізнавання символів – як це працює [Електронний ресурс]: <https://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works/>
2. Як отримати найвищі результати розпізнавання [Електронний ресурс]: <http://www.dlib.org/dlib/march09/holley/03holley.html>
3. Що таке екстракція даних [Електронний ресурс]:<https://www.hyland.com/en/resources/terminology/data-capture/what-is-data-capture>
4. ABBYY FlexiCapture [Електронний ресурс]:
<https://processflows.co.uk/direct/process-automation-components/data-capture/data-capture-software/abbyy-software/abbyy-flexicapture/>
5. Top OCR software [Електронний ресурс]:
<https://web.archive.org/web/20170223213719/http://ocrworld.com/software/5-in-depth/149-top-ocr-software.html>
6. Документація ABBYY FlexiCapture [Електронний ресурс]:
https://help.abbyy.com/en-us/flexicapture/12/developer/introduction_ag
7. 7. Документація MAMP [Електронний ресурс] :
<https://www.mamp.info/en/mamp-pro/mac/>

					<i>КНТЕУ 121 06-08.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.			24.01.20	Розробка інформаційної системи оптичного розпізнавання тексту й екстракції слабоструктурованих	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник	Гнатченко Д.Д.			24.01.20		<i>СВД</i>	<i>50</i>	<i>52</i>
Гарант	Криворучко О.В.			24.01.20		Факультет інформаційних технологій 2м курс, 6 група		
Розробив	Дьяченко М.І.			24.01.20				
					<i>Список використаних джерел</i>			

ТЕХНІЧНЕ ЗАВДАННЯ

Однією з найголовніших заporук відповідності вимогам та високої якості розробленого програмного рішення є грамотно спроектоване до початку розробки технічне завдання. Отже, далі розкриємо невід'ємні положення технічного завдання до розробки інформаційної системи у вигляді веб-додатку із використанням технології оптичного розпізнавання символів та екстракції слабоструктурованих даних.

1. Загальні відомості

Назва проекту: EasyLab.

Планові строки розробки: серпень 2019 – вересень 2019.

Потенційні користувачі: студенти та викладачі навчальних закладів.

Призначення програмного рішення: автоматизувати процес валідації та здачі лабораторних робіт, створити систему для зберігання завдань і забезпечення їх доступності.

Мета створення програмного рішення: опанування сучасних технологій розробки інформаційних та їх закріплення на практиці.

2. Структура проекту

Оскільки процес здачі робіт передбачає наявність централізованого сервера і інтерфейсу, програмне рішення складатиметься з таких частин: серверний код, клієнтський код, база даних, модуль розпізнавання та екстракції даних.

Серверна частина виконуватиметься на ПК, оскільки він надає доступ до своєї системи і бази даних.

					<i>КНТЕУ 121 06-08.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Розробка інформаційної системи оптичного розпізнавання тексту й екстракції слабоструктурованих даних	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		28.02.20		<i>ТЗ</i>	<i>51</i>	<i>52</i>
Керівник		Гнатченко Д.Д.		28.02.20		Факультет інформаційних технологій 2м курс, 6 група		
Гарант		Криворучко О.В.		28.02.20				
Розробив		Дьяченко М.І.		28.02.20	<i>Технічне завдання</i>			

Клієнтська частина буде реалізована у вигляді HTML сторінок, що відображатимуться у браузері.

3. Функціональні вимоги до додатку

- Користувачі система належать до однієї з трьох ролей – викладач, староста, студент.
- Адміністратор додає в систему викладачів, групи та їх старост.
- Викладач додає в систему дисципліни та завдання до лабораторних робіт, після здачі робіт викладач виставляє оцінку.
- Староста додає в систему студентів
- Студенти мають доступ до завантаження завдань з серверу та завантаження виконаних робіт на сервер.

					<i>КНТЕУ 121 06-08.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		52

ДОДАТКИ

Додаток А

Код серверної частини інформаційної системи:

```
taskUploader.php
<?php

session_start();

$message = "";
if (isset($_POST['uploadBtn']) && $_POST['uploadBtn']
== 'Upload')
{
    if (isset($_FILES['uploadedFile']) &&
$_FILES['uploadedFile']['error'] === UPLOAD_ERR_OK)
    {
        // get details of the uploaded file
        $fileTmpPath = $_FILES['uploadedFile']['tmp_name'];
        $fileName = $_FILES['uploadedFile']['name'];
        $fileSize = $_FILES['uploadedFile']['size'];
        $fileType = $_FILES['uploadedFile']['type'];
        $fileNameCmps = explode(".", $fileName);
        $fileExtension = strtolower(end($fileNameCmps));

        // sanitize file-name
        $newFileName = md5(time() . $fileName) . '.' .

$fileExtension;

        // check if file has one of the following extensions
        $allowedfileExtensions = array('jpg', 'gif', 'png', 'zip',
'txt', 'xls', 'doc', 'docx', 'pdf');
```

```

if (in_array($fileExtension, $allowedfileExtensions))
{
    // directory in which the uploaded file will be moved
    $uploadFileDir = './uploaded_files/tasks';
    $dest_path = $uploadFileDir . $newFileName;

    if(move_uploaded_file($fileTmpPath, $dest_path))
    {
        $message ='File is successfully uploaded.';
    }
    else
    {
        $message = 'There was some error moving the file to
upload directory. Please make sure the upload directory is writable by web server.';
    }
}
else
{
    $message = 'Upload failed. Allowed file types: ' .
implode(',', $allowedfileExtensions);
}
}
else
{
    $message = 'There is some error in the file upload.
Please check the following error.<br>';
    $message .= 'Error:' . $_FILES['uploadedFile']['error'];
}
}

```

```
$_SESSION['message'] = $message;
```

```
function getInputTaskNumber () {  
    $inputTaskNumber = $_POST['taskNumber'];  
    return $inputTaskNumber;  
}
```

```
function getInputSubject () {  
    $inputSubject = $_POST['subject'];  
    return $inputSubject;  
}
```

```
function addTaskToDb($subject, $taskNumber,  
$filePath) {
```

```
    DEFINE('DB_USERNAME', 'root');  
    DEFINE('DB_PASSWORD', 'root');  
    DEFINE('DB_HOST', 'localhost');  
    DEFINE('DB_DATABASE', 'mybase');
```

```
    $mysqli = new mysqli(DB_HOST,  
DB_USERNAME, DB_PASSWORD, DB_DATABASE);
```

```
    if (mysqli_connect_error()) {  
        die('Connect Error ('.mysqli_connect_errno().')  
' . mysqli_connect_error());  
    }
```

```

        $subjectIdResult = $mysqli->query("SELECT
        `id` FROM `subjects` WHERE name = '". $subject.'");

        function getSubjectId ($result_set) {
            while(($row = $result_set->fetch_assoc())
            != false){
                return $row['id'];
            }
        }

        $subjectId = getSubjectId($subjectIdResult);
        $mysqli->query("INSERT INTO `tasks`
        (`subjectId`, `taskNumber`, `filePath`) VALUES ('". $subjectId."",
        '". $taskNumber."', '". $filePath.'");

        $mysqli->close();
    }
    $subject = getInputSubject();
    $taskNumber = getInputTaskNumber();

    addTaskToDb($subject, $taskNumber, $dest_path);

    header("Location: teacher.php");

```

?>

fileProcessor.php

<?php

```

session_start();

$message = "";
if (isset($_POST['uploadBtn']) && $_POST['uploadBtn']
== 'Upload')
{
    if (isset($_FILES['uploadedFile']) &&
$_FILES['uploadedFile']['error'] === UPLOAD_ERR_OK)
    {
        // get details of the uploaded file
        $fileTmpPath = $_FILES['uploadedFile']['tmp_name'];
        $fileName = $_FILES['uploadedFile']['name'];
        $fileSize = $_FILES['uploadedFile']['size'];
        $fileType = $_FILES['uploadedFile']['type'];
        $fileNameCmps = explode(".", $fileName);
        $fileExtension = strtolower(end($fileNameCmps));

        // sanitize file-name
        $newFileName = md5(time() . $fileName) . '.' .

$fileExtension;

        // check if file has one of the following extensions
        $allowedfileExtensions = array('jpg', 'gif', 'png', 'zip',
'txt', 'xls', 'doc', 'docx');

        if (in_array($fileExtension, $allowedfileExtensions))
        {
            // directory in which the uploaded file will be moved
            $uploadFileDir = './uploaded_files/';
            $dest_path = $uploadFileDir . $newFileName;

```

```

        if(move_uploaded_file($fileTmpPath, $dest_path))
        {
            $message ='File is successfully uploaded.';
        }
        else
        {
            $message = 'There was some error moving the file to
upload directory. Please make sure the upload directory is writable by web server.';
        }
    }
    else
    {
        $message = 'Upload failed. Allowed file types: ' .
implode(',', $allowedfileExtensions);
    }
}
else
{
    $message = 'There is some error in the file upload.
Please check the following error.<br>';
    $message .= 'Error:' . $_FILES['uploadedFile']['error'];
}
}
$_SESSION['message'] = $message;
header("Location: index.php");

function openSession () {
    $curl = curl_init();
    curl_setopt_array($curl, array(

```

```

CURLOPT_URL =>
"https://usa.flexicapture.com/FlexiCapture12/Server/FCAuth/API/v1/JSON?tenant
=abbyy_AmericasSupport",

CURLOPT_RETURNTRANSFER => true,
CURLOPT_ENCODING => "",
CURLOPT_MAXREDIRS => 10,
CURLOPT_TIMEOUT => 0,
CURLOPT_FOLLOWLOCATION => true,
CURLOPT_HTTP_VERSION =>
CURL_HTTP_VERSION_1_1,
CURLOPT_CUSTOMREQUEST => "POST",
CURLOPT_POSTFIELDS
=> "{\r\n\t\"MethodName\": \"OpenSession\", \r\n\t\"Params\": {\r\n\t\t\"roleType\": 6
, \r\n\t\t\"stationType\": 3\r\n\t}\r\n}",

CURLOPT_HTTPHEADER => array(
    "Content-Type: application/json",
    "Authorization: Basic
YXBpX3VzZXI6cGFzc3dvcmQx",
    "Cookie:
FlexiCaptureTmpPrn=Ticket=CDF283C6EE9221C65D7DED31E8DFDB7461C99
FEB1D3D8C7491B9288D7BECCF3DCE6DFE343E7FF9F6AD008CC78AD85D
EF4EE950C5E14FA81107D4E8A4F1077B9B12CE7B4A2B2A1A793A4DB95C
DB990FF6"
),
));
$response = curl_exec($curl);
curl_close($curl);
$arr = json_decode($response);
return $sessionId = $arr->sessionId;
}

```



```
"Content-Type: application/json",  
"Authorization: Basic  
YXBpX3VzZXI6cGFzc3dvcmQx",  
"Cookie:  
FlexiCaptureTmpPrn=Ticket=CDF283C6EE9221C65D7DED31E8DFDB7461C99  
FEB1D3D8C7491B9288D7BECCF3D82E22C079F1FFBF2E11358C8042886715  
3BD22EEE445D7FEAE665A434F2C0E0C97B059E94D52923319DFA9EA91E9  
2765"
```

```
),  
));  
$response = curl_exec($curl);  
curl_close($curl);  
$sarr = json_decode($response);  
return $batchId = $sarr->batchId;  
}
```

```
function openBatch ($sessionId, $batchId) {  
    $curl = curl_init();  
    curl_setopt_array($curl, array(  
        CURLOPT_URL =>  
        "https://usa.flexicapture.com/FlexiCapture12/Server/FCAuth/API/v1/JSON?tenant  
        =abbyy_AmericasSupport",  
        CURLOPT_RETURNTRANSFER => true,  
        CURLOPT_ENCODING => "",  
        CURLOPT_MAXREDIRS => 10,  
        CURLOPT_TIMEOUT => 0,  
        CURLOPT_FOLLOWLOCATION => true,
```

```

CURLOPT_HTTP_VERSION =>
CURL_HTTP_VERSION_1_1,
CURLOPT_CUSTOMREQUEST => "POST",
CURLOPT_POSTFIELDS
=> "{\r\n\t\"MethodName\": \"OpenBatch\", \r\n\t\"Params\": {\r\n\t\t\"sessionId\": \"$
sessionId.\", \r\n\t\t\"batchId\": \"$batchId.\" \r\n\t\t\r\n\t}\r\n\r\n}",
CURLOPT_HTTPHEADER => array(
    "Content-Type: application/json",
    "Authorization: Basic
YXBpX3VzZXI6cGFzc3dvcmQx",
    "Cookie:
FlexiCaptureTmpPrn=Ticket=CDF283C6EE9221C65D7DED31E8DFDB7461C99
FEB1D3D8C7491B9288D7BECCF3DD57BB49028B1789C26577C5C005E57E8
04ADCF92D2AD18DA19C7492C9D9FD87A25A133779C28B7184C0F0AFDA
E9DD414"
),
));
$response = curl_exec($curl);

curl_close($curl);
echo $response;
}

function addNewDocument ($sessionId, $batchId,
$documentBytes) {

    $curl = curl_init();

```


"Authorization: Basic

YXBpX3VzZXI6cGFzc3dvcmQx",

"Cookie:

FlexiCaptureTmpPrn=Ticket=CDF283C6EE9221C65D7DED31E8DFDB7461C99
FEB1D3D8C7491B9288D7BECCF3D2869E1ECD36FA3E0119522E31D3BB9E
8ADB7F5AED56BB1A567F0A366852CEC97FF7CA73CDAB37D9D00C0A741
FB266741"

),

));

\$response = curl_exec(\$curl);

\$sarr = json_decode(\$response);

return \$documentId = \$sarr->documentId;

curl_close(\$curl);

}

function closeBatch (\$sessionId, \$batchId) {

\$curl = curl_init();

curl_setopt_array(\$curl, array(

CURLOPT_URL =>

"https://usa.flexicapture.com/FlexiCapture12/Server/FCAuth/API/v1/JSON?tenant
=abbyy_AmericasSupport",

CURLOPT_RETURNTRANSFER => true,

CURLOPT_ENCODING => "",

CURLOPT_MAXREDIRS => 10,

```

CURLOPT_TIMEOUT => 0,
CURLOPT_FOLLOWLOCATION => true,
CURLOPT_HTTP_VERSION =>
CURL_HTTP_VERSION_1_1,
CURLOPT_CUSTOMREQUEST => "POST",
CURLOPT_POSTFIELDS => "{\r\n
\"MethodName\": \"CloseBatch\", \r\n  \"Params\": {\r\n
\"sessionId\": \".$sessionId.\", \r\n    \"batchId\": \".$batchId.\" \r\n    }\r\n}",
CURLOPT_HTTPHEADER => array(
    "Content-Type: application/json",
    "Authorization: Basic
YXBpX3VzZXI6cGFzc3dvcmQx",
    "Cookie:
FlexiCaptureTmpPrn=Ticket=CDF283C6EE9221C65D7DED31E8DFDB7461C99
FEB1D3D8C7491B9288D7BECCF3D4D87B53ED51F3C437A2DB2C9FD5ED1
744B268AC98E7DAE4AF928ED29A6B61E0E35041354A4C84C92FFCF13B42
8186AF1"
),
);

$response = curl_exec($curl);

curl_close($curl);
echo $response;
}

function processBatch ($sessionId, $batchId) {

    $curl = curl_init();

```

```

curl_setopt_array($curl, array(
    CURLOPT_URL =>
"https://usa.flexicapture.com/FlexiCapture12/Server/FCAuth/API/v1/JSON?tenant
=abbyy_AmericasSupport",

    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 0,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_HTTP_VERSION =>
CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "POST",
    CURLOPT_POSTFIELDS => "{\r\n
\"MethodName\": \"ProcessBatch\", \r\n  \"Params\": {\r\n
\"sessionId\": \".$sessionId.\", \r\n    \"batchId\": \".$batchId.\" \r\n    \r\n  }\r\n}",
    CURLOPT_HTTPHEADER => array(
        "Content-Type: application/json",
        "Authorization: Basic
YXBpX3VzZXI6cGFzc3dvcmQx",
        "Cookie:
FlexiCaptureTmpPrn=Ticket=CDF283C6EE9221C65D7DED31E8DFDB7461C99
FEB1D3D8C7491B9288D7BECCF3D551AEF4A1B4741B0A0BA143A1CD51D
26B388B11F0D43CCEB3C6F869B26CB18CD62C6325C5E0BB96EFF81C756B
4A66189"
    ),
));

$response = curl_exec($curl);

curl_close($curl);

```

```

    }

    function getResultsList ($sessionId, $batchId,
        $documentId) {

        $curl = curl_init();

        curl_setopt_array($curl, array(
            CURLOPT_URL =>
                "https://usa.flexicapture.com/FlexiCapture12/Server/FCAuth/API/v1/JSON?tenant
                =abbyy_AmericasSupport",
            CURLOPT_RETURNTRANSFER => true,
            CURLOPT_ENCODING => "",
            CURLOPT_MAXREDIRS => 10,
            CURLOPT_TIMEOUT => 0,
            CURLOPT_FOLLOWLOCATION => true,
            CURLOPT_HTTP_VERSION =>
                CURL_HTTP_VERSION_1_1,
            CURLOPT_CUSTOMREQUEST => "POST",
            CURLOPT_POSTFIELDS => "{\n
                \"MethodName\": \"GetDocumentResultsList\", \n
                \"Params\": {\n
                \"sessionId\": \".$sessionId.\", \n
                \"batchId\": \".$batchId.\", \n
                \"documentId\":
                \".$2'.\" \n
                }\n
                }",
            CURLOPT_HTTPHEADER => array(
                "Authorization: Basic
                YXBpX3VzZXI6cGFzc3dvcmQx",
                "Content-Type: application/json",
                "Cookie:
                FlexiCaptureTmpPrn=Ticket=CDF283C6EE9221C65D7DED31E8DFDB7461C99
                FEB1D3D8C7491B9288D7BECCF3DB8A83A6BD189924F888ECA36AC9951B

```

019A99C81BDDDB5465DA502EA017B4942D099E285F0CBA865B8CD09D56D
239FAFF"

),
));

\$response = curl_exec(\$curl);

curl_close(\$curl);

\$arr = json_decode(\$response, true);

return \$arr;

}

function loadDocumentResult (\$sessionId, \$batchId,

\$fileName, \$documentId) {

\$curl = curl_init();

curl_setopt_array(\$curl, array(

CURLOPT_URL =>

"https://usa.flexicapture.com/FlexiCapture12/Server/FCAuth/API/v1/JSON?tenant
=abbyy_AmericasSupport",

CURLOPT_RETURNTRANSFER => true,

CURLOPT_ENCODING => "",

CURLOPT_MAXREDIRS => 10,

CURLOPT_TIMEOUT => 0,

CURLOPT_FOLLOWLOCATION =>

true,

```

CURLOPT_HTTP_VERSION =>
CURL_HTTP_VERSION_1_1,

CURLOPT_CUSTOMREQUEST =>

"POST",

CURLOPT_POSTFIELDS => "{\n
  \"MethodName\": \"LoadDocumentResult\", \n  \"Params\": {\n
    \"sessionId\": \".$sessionId.\", \n    \"batchId\": \".$batchId.\", \n    \"documentId\":
    \".$documentId.\", \n    \"fileName\": \"\".$fileName.\"\" \n  } \n}",

CURLOPT_HTTPHEADER => array(
  "Authorization: Basic
  YXBpX3VzZXI6cGFzc3dvcmQx",

  "Content-Type: application/json",

  "Cookie:
  FlexiCaptureTmpPrn=Ticket=CDF283C6EE9221C65D7DED31E8DFDB7461C99
  FEB1D3D8C7491B9288D7BECCF3D4F367D99F6FA4749C5F35C700872ECCA
  0AFA64FFE04773F6E2453BFA1E160361CCE039B02A3439A80266BFA0C341
  B5E4"

),
);

$response = curl_exec($curl);

curl_close($curl);

$arr = json_decode($response, true);

return $arr;

}

$sessionId = openSession();

```

```

$batchId = addNewBatch($sessionId);
// //echo $batchId;
openBatch($sessionId, $batchId);
$file = file_get_contents($dest_path);
$documentBytes = base64_encode($file);
$documentId = addNewDocument($sessionId, $batchId,
$documentBytes);

closeBatch($sessionId, $batchId);
processBatch($sessionId, $batchId);
echo 'session: '. $sessionId.'
document'.'$documentId.'batch: '.$batchId;

sleep(30);
$fileName = getResultsList($sessionId, $batchId,
$documentId);

print_r($fileName);
$fileNameToDownload = $fileName['fileNames'][0];
echo $fileNameToDownload;

$fileNameToDownload = str_replace('\\', '\\\\\\\\',
$fileNameToDownload);

echo $fileNameToDownload;

$resultByteArray = loadDocumentResult($sessionId,
$batchId, "Export\\\\TESTbatch\\\\Data_00000002.json", 2);
print_r($resultByteArray) ;
$resultBytes = $resultByteArray['file']['Bytes'];
$decodedJson = base64_decode($resultBytes);

file_put_contents('Export\\TESTbatch\\Data_00000002.json', $decodedJson);

```

```

$string =
file_get_contents('Export\TESTbatch\Data_00000002.json');

$json_a = json_decode($string, true);

print_r($json_a);

echo "<br><br>";

$slabNumber =
$json_a['Documents'][0]['DocumentData']['Fields'][0]['Value'];
$studentName =
$json_a['Documents'][0]['DocumentData']['Fields'][1]['Value'];
$stopic =
$json_a['Documents'][0]['DocumentData']['Fields'][2]['Value'];

function addLabToDb($student, $stopic, $slabNumber,
$filePath) {

    DEFINE('DB_USERNAME', 'root');
    DEFINE('DB_PASSWORD', 'root');
    DEFINE('DB_HOST', 'localhost');
    DEFINE('DB_DATABASE', 'mybase');

    $mysqli = new mysqli(DB_HOST,
DB_USERNAME, DB_PASSWORD, DB_DATABASE);

```

```

        if (mysqli_connect_error()) {
            die('Connect Error ('.mysqli_connect_errno().')
'.mysqli_connect_error());
        }

        $mysqli->query("INSERT INTO `labs` (`student`,
`topic`, `labNumber`, `filePath`) VALUES ('.$student.', '$topic.',
'$.labNumber.', '$filePath.')");

        $mysqli->close();
    }

```

```
addLabToDb($studentName,$topic,$labNumber,$dest_path);
```

```
?>
```

```
Admin_panel.php
```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" href="main_style.css">
    <link rel="stylesheet" href="normalize.css">
    <title>EasyLab</title>

```

```

</head>
<body>
    <section class="main">
        <a href="signin.php" class="sign_out"><button
class="sign_out" style="background-color: #F4FDC0;">Вихід</button></a>
        <p>Привіт, Адміністратор!
        Нижче ти можеш побачити таблиці з даними які ти можеш
        редагувати.</p>

```

```

<?php
function getResult ($result_set) {
    while(($row = $result_set->fetch_assoc()) != false){

        echo $row[id]. ' '.$row[login];
        echo "<br>";
    }
}

function getUsers () {

    DEFINE('DB_USERNAME', 'root');
    DEFINE('DB_PASSWORD', 'root');
    DEFINE('DB_HOST', 'localhost');
    DEFINE('DB_DATABASE', 'mybase');

    $mysqli = new mysqli(DB_HOST, DB_USERNAME,
    DB_PASSWORD, DB_DATABASE);

```

```

        if (mysqli_connect_error()) {
            die('Connect Error ('.mysqli_connect_errno().')
'.mysqli_connect_error());
        }

        $result = $mysqli->query("SELECT `id`, `login` FROM
`users` ");

        $rows = mysqli_num_rows($result); // количество
полученных строк

        echo "<table><p class='\"tb_name\">Викладачі</p>";
        for ($i = 0 ; $i < $rows ; ++$i)
        {
            $row = mysqli_fetch_row($result);
            echo "<tr>";
            for ($j = 0 ; $j < 3 ; ++$j) echo "<td>$row[$j]</td>";
            echo "<td><a><button class='\"table_button
pass_reset\">Відновити пароль</button></a></td><td><a><button
class='\"table_button delete\">Видалити</button></a></td>\".</tr>";

        }

        echo "<tr class='\"ad\"><td><a
href='\"add_professor.html\"><button class='\"add\">+</button></a></td></tr>";
        echo "</table>";

        // очищаем результат
        mysqli_free_result($result);

        $mysqli->close();

```

```

        //return getResult($result);
    }

    getUsers();

?>

```

```

<table class="groups">
    <p class="tb_name">Групи</p>
    <tr><td class="number">1.</td><td
class="gr_name">БІКК</td><td><a><button class="table_button
delete">Видалити</button></a></td></tr>
    <tr class="ad"><td><a href="add_group.html"><button
class="add">+</button></a></td></tr>
</table>

</section>

</body>

</html>

```

Authenticator.php

```

<?php
    DEFINE('DB_USERNAME', 'root');
    DEFINE('DB_PASSWORD', 'root');
    DEFINE('DB_HOST', 'localhost');
    DEFINE('DB_DATABASE', 'mybase');

    function getInputLogin () {
        $inputLogin = $_POST['email'];
    }

```

```

        return $inputLogin;
    }

    function getLogin ($result_set) {
        while(($row = $result_set->fetch_assoc()) != false){
            return $row['login'];
        }
    }

    function isHeadman ($result_set) {
        while(($row = $result_set->fetch_assoc()) != false){
            return $row['isHeadman'];
        }
    }

    function isTeacher ($result_set) {
        while(($row = $result_set->fetch_assoc()) != false){
            return $row['isTeacher'];
        }
    }

    function getUser () {

        $mysqli = new mysqli(DB_HOST, DB_USERNAME,
DB_PASSWORD, DB_DATABASE);

```

```

        if (mysqli_connect_error()) {
            die('Connect Error ('.mysqli_connect_errno().')
'.mysqli_connect_error());
        }

        $result = $mysqli->query("SELECT `login`, `isHeadman`,
`isTeacher` FROM `users` where `login` = '".getInputLogin()."'");
        $mysqli->close();
        return $result;
    }

```

```

    $user = getUser();
    $login = getLogin($user);

```

```

    if ($login == null) {
        header("Location: user_not_exist.php");
    }

```

```

    if ($user != null) {
        $user = getUser();
        $isHeadman = isHeadman($user);
        if ($isHeadman == 1) {
            header("Location: headman.php");
        }
        $user = getUser();
        $isTeacher = isTeacher($user);
        if ($isTeacher == 1) {
            header("Location: teacher.php");
        } else {

```

```

        header("Location: student.php");
    }
}

?>

```

addTask.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet" href="main_style.css">
    <link rel="stylesheet" href="normalize.css">
    <title>EasyLab</title>
</head>
<body>
    <section class="main">
        <a href="#" class="sign_out" style=""><button
class="sign_out" style="background-color: #F4FDC0;">Вихід</button></a>
        <div class="form">
            <form method="POST" action="taskUploader.php"
enctype="multipart/form-data" style="height:480px;">
                <label>Номер роботи</label>
                <input type="text" name="taskNumber">
                <label>Дисципліна</label>

```

```

<?php

function getSubjects () {

    DEFINE('DB_USERNAME',
'root');

    DEFINE('DB_PASSWORD',
'root');

    DEFINE('DB_HOST',
'localhost');

    DEFINE('DB_DATABASE',
'mybase');

    $mysqli = new
mysqli(DB_HOST, DB_USERNAME, DB_PASSWORD, DB_DATABASE);

    if (mysqli_connect_error()) {
        die('Connect Error
(' .mysqli_connect_errno().') '.mysqli_connect_error());
    }

    $result = $mysqli-
>query("SELECT `name` FROM `subjects` ");

    $rows =
mysqli_num_rows($result); // количество полученных строк

    echo "<select
name=\"subject\" id=\"\">";

```

```

        for ($i = 0 ; $i < $rows ; ++$i)
        {
            $row =
mysqli_fetch_row($result);

            for ($j = 0 ; $j < 3 ;
++$j) echo "<option value=\"{$row[$j]}\">{$row[$j]}</option>";

        }
        echo "</select>";

        // очищаємо результат
        mysqli_free_result($result);

        $mysqli->close();
        //return getResult($result);
    }

```

getSubjects();

?>

<!-- <div class="dropzone">

<p style="width: 200px;">Виберіть

локальний файл або перетягніть його в цю зону

</div>

<a href="" class="" style="margin-top: 40px;

width: 196px;"><button class="enter" style="background-color:

#cdf2f4;">Додати завдання</button> -->

<div>

Upload a File:

```
<input type="file" name="uploadedFile" />
</div>

<input type="submit" name="uploadBtn"
value="Upload" />

</form>
</div>
</section>
</body>
</html>
```

addTeacher.php

```
<?php
```

```
function getInputTeacherEmail (){
    $inputTeacherEmail = $_POST['email'];
    return $inputTeacherEmail;
}
```

```
function getInputTeacherName (){
    $inputTeacherName = $_POST['name'];
    return $inputTeacherName;
}
```

```
function addTeacherToDb($name, $email) {
```

```
    DEFINE('DB_USERNAME', 'root');
```

```
    DEFINE('DB_PASSWORD', 'root');
```

```

        DEFINE('DB_HOST', 'localhost');
        DEFINE('DB_DATABASE', 'mybase');

        $mysqli = new mysqli(DB_HOST, DB_USERNAME,
        DB_PASSWORD, DB_DATABASE);

        if (mysqli_connect_error()) {
            die('Connect Error ('.mysqli_connect_errno().')
            '.mysqli_connect_error());
        }

        $mysqli->query("INSERT INTO `users` (`login`, `name`,
        `password`, `reg_date`) VALUES ('".$email."', '".$name."', MD5('123123'),
        UNIX_TIMESTAMP())");

        $mysqli->close();
    }

    $inputName = getInputTeacherName();
    $inputEmail = getInputTeacherEmail();

    addTeacherToDb($inputName, $inputEmail);
    header ("Location: admin_panel.php");

    ?>

```