GENERATIVE AI FOR TRADITIONAL FORM CONVERTER

MR. PRAPAKORN BUTYOJANTO
MR. PHANASORN SRISAYAM
MR. NATTAWUT PIANOK

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)
FACULTY OF ENGINEERING
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI
2024

Generative AI for Traditional Form Converter

Mr. PRAPAKORN BUTYOJANTO
Mr. PHANASORN SRISAYAM
Mr. NATTAWUT PIANOK

A Project Submitted in Partial Fulfillment
of the Requirements for
the Degree of Bachelor of Engineering (Computer Engineering)
Faculty of Engineering
King Mongkut's University of Technology Thonburi
2024

Project Committee

. . . . Project Advisor
(Dr. Kittipong Piyawanno , Ph.D.)

. . . . Committee Member
(Assoc.Prof. Natasha Dejdumrong, D.Tech.Sci.)

. . . . Committee Member
(Assoc.Prof. Peerapon Siripongwutikorn, Ph.D.)

. . . . Committee Member
(Naveed Sultan)

Copyright reserved

Project Title	Generative AI for Traditional Form Converter
Credits	3
Member(s)	Mr. PRAPAKORN BUTYOJANTO Mr. PHANASORN SRISAYAM Mr. NATTAWUT PIANOK
Project Advisor	Dr. Kittipong Piyawanno , Ph.D.
Program	Bachelor of Engineering
Field of Study	Computer Engineering
Department	Computer Engineering
Faculty	Engineering
Academic Year	2024

Abstract

Generative AI for Traditional Form Converter is a project developed through a web application called PaperlessTransform Application to solve the problem of taking a long time to convert paper forms into web applications. The development of this web application uses artificial intelligence to analyze the types of data of questions.

As the demand for form conversion increases, system developers need to analyze forms and design database systems, design web application pages, and develop new systems. This results in increased work time. In addition, system developers face increased workloads, causing personnel to spend inefficient time on their work. Our project focuses on developing a web application that can convert documents in the form of paper forms or electronic files into web application formats. It uses optical character recognition techniques to convert text images into text formats to process the text from the converted text images to detect questions in the form format.

The project team focuses on developing a web application that has the ability to detect questions and store web form data. The aim is to reduce the burden on system developers. The results after testing the web application in the work show that the web application can detect questions in the form and store data at a satisfactory level. Therefore, it can be concluded that the project can significantly solve the problem of increased work time for system developers.

Keywords: Web Application / Optical Character Recognition (OCR) / Database Design

หัวข้อปริญญานิพนธ์	เว็บแอปพลิเคชัน AI สำหรับการแปลงฟอร์มกระดาษเป็นเว็บฟอร์ม
หน่วยกิต	3
ผู้เขียน	นายประภากร บุตรโยจันโท นายพณศร ศรีสยาม นายณัฐวุฒิ เพียนอก
อาจารย์ที่ปรึกษา	ดร. กิตติพงษ์ ปิยะวรรณโณ
หลักสูตร	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมคอมพิวเตอร์
คณะ	วิศวกรรมศาสตร์
ปีการศึกษา	2567

บทคัดย่อ

Generative AI for Traditional Form Converter เป็นโครงการที่จัดทำขึ้นผ่านการพัฒนาผ่านเว็บแอปพลิเคชั่นในชื่อ PaperlessTransform Application เพื่อแก้ไขปัญหาการใช้ระยะเวลานานในการแปลงแบบฟอร์มกระดาษเป็นรูปแบบเว็บแอปพลิเคชัน โดยการพัฒนาเว็บแอปพลิเคชันนี้ได้มีการใช้ประยุกต์ใช้ปัญญาประดิษฐ์สำหรับการวิเคราะห์เกี่ยวกับประเภทของข้อมูลของคำถาม ตามความต้องการที่เพิ่มขึ้นของการแปลงแบบฟอร์ม ดังนั้นนักพัฒนาระบบจึงจำเป็นต้องวิเคราะห์แบบฟอร์มและออกแบบระบบฐานข้อมูลพร้อมทั้งการออกแบบหน้าเว็บแอปพลิเคชัน รวมไปถึงการพัฒนาระบบขึ้นมาใหม่ จึงส่งผลให้ต้องใช้ระยะเวลาในการทำงานที่เพิ่มขึ้น นอกจากนี้ นักพัฒนาระบบต้องเผชิญกับปัญหาภาระงานที่มากขึ้น ส่งผลให้บุคคลากรใช้เวลาในการทำงานอย่างไม่มีประสิทธิภาพ โดยโครงการของเรามุ่งเน้นการพัฒนาเว็บแอปพลิเคชันที่สามารถแปลงเอกสารในรูปแบบของฟอร์มกระดาษ หรือ ไฟล์อิเล็กทรอนิกส์ให้เป็นรูปแบบของเว็บแอปพลิเคชัน โดยใช้เทคนิคการรู้จดจำอักขระด้วยแสงในการแปลงภาพข้อความให้เป็นรูปแบบข้อความเพื่อนำข้อความดังกล่าวจากการแปลงภาพข้อความนำมาประมวลผลในการตรวจจับคำถามในรูปแบบฟอร์ม ทางคณะผู้จัดทำโครงการมีการเน้นการพัฒนาเว็บแอปพลิเคชันที่มีความสามารถในการตรวจจับคำถามและความสามารถในการเก็บข้อมูลของเว็บฟอร์ม โดยมีวัตถุประสงค์เพื่อลดภาระของนักพัฒนาระบบ โดยผลลัพธ์หลังจากมีการทดลองใช้เว็บแอปพลิเคชั่นดังกล่าวในการทำงานแสดงให้เห็นว่าเว็บแอปพลิเคชันสามารถตรวจจับคำถามในแบบฟอร์มและเก็บข้อมูลได้ในระดับที่น่าพึ่งพอใจ ดังนั้นสรุปได้ว่าโครงการสามารถแก้ไขปัญหาการใช้ระยะเวลาในการทำงานที่เพิ่มขึ้น ของนักพัฒนาระบบได้อย่างมีนัยสำคัญ

คำสำคัญ: เว็บแอปพลิเคชัน / การรู้จดจำอักขระด้วยแสง / ออกแบบระบบฐานข้อมูล

ACKNOWLEDGMENTS

The authors would like to express special sincere gratitude to Dr. Kittipong Piyawanno, the project advisor, who always supported and guided the project's direction throughout this project journey. His expertise and mentorship have played an important role in our project to shape the project.

Additionally, I would like to thank all the contributors across various platforms, such as Medium, Stack Overflow, and ChatGPT, whose shared knowledge and expertise have been invaluable. Their contributions have significantly enhanced my understanding and helped us refine our work.

Finally, we would like to express our thanks to everyone, including our family, friends, and everyone who has contributed and supported this project. All the support and encouragement have been integral to its successful completion.

CONTENTS

PAGE

CHAPTER

LIST OF TABLES

TABLE **PAGE**

LIST OF FIGURES

FIGURE **PAGE**

LIST OF SYMBOLS

SYMBOL		**UNIT**
α	Test variable	m^2
λ	Interarival rate	jobs/ second
μ	Service rate	jobs/ second

LIST OF TECHNICAL VOCABULARY AND ABBREVATIONS

ABC	=	Adaptive Bandwidth Control
MANET	=	Mobile Ad Hoc Network
Test	=	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam non condimentum purus. Pellentesque sed augue sapien. In volutpat quis diam laoreet suscipit. Curabitur fringilla sem nisi, at condimentum lectus consequat vitae.

CHAPTER 1 INTRODUCTION

1.1 Problem Statement

««« Updated upstream Nowadays, there is a shift towards digitalization, whether it is using electronic devices to record instead of using paper or storing data in a database instead of recording on paper. However, there are many things that have not been transformed to be more digital, such as official documents and business documents that are still paper or forms that have been recorded on paper before. For things that have not been developed to be more digital, we are especially interested in improving the filling of paper forms. We found that the steps in the development process that change from paper forms to web forms require developers to analyze forms, create databases, and develop web applications, which takes a lot of time and the number of developers. We see that converting paper forms to digital forms by taking pictures of the forms and the system will create a web form for them. Improving the filling of paper forms to be more digital will help reduce global warming caused by excessive paper use.

1.2 Potential Benefits

Potential Benefitsof this web application include improved efficiency through automated form creation, enabling the rapid digitization of paper forms and saving valuable time that can be used for other tasks, and potentially reducing costs by eliminating unnecessary labor in form processing.

======= In recent year, the world has become more digitized than ever, whether it be using electronic devices for note-taking instead of paper or storing a data in a database rather than use a paper documentation, However, there are still many aspects that have not been transform to be a digital, such as a government organization that still rely on paper documentation or the data forms that have been previously recorded on paper. However, for those that have not been developed to be digital, we were motivated to improve the efficiency of form filling. And we found that the development process from paper-based forms to web forms required developers to do an analysis of the form, create a database, and develop a front-end and backend system. Which causes a developer time to spend and resources of developers. We acknowledge this difficulty and offer a solution that converts paper forms into web form ones so that the system can automatically create a web form by just taking a picture of the form. By decreasing excessive paper use, this development in digitizing form-filling procedures will not only increase efficiency but also lessen the environmental impact and help to slow down global warming. »»»> Stashed changes

1.3 Objectives

- To reduce the workload and development process for developers.
- To acquire the knowledge and skills necessary for developing an AI-powered web application
- To acquire proficiency in utilizing a Large language models and adapt its capabilities to suit the requirements of this project.
- To be the secure all data and form management website

1.4 Scope of Work

The scope of this project involves the development of a web application that enables users to upload a PDF or image file. The web application will process text extraction using optical character recognition (OCR). The primary function of the web application includes creating a form, editing a form, deleting a form, and filling a form. The final deliverable of this project will be a responsive web interface web application that allows users to manage the form and view the data, including ensuring data privacy and security measures to protect sensitive information by implementing authentication for the creator and a normal user. The project involves research of optical character recognition (OCR) for the text extraction from the image and the development of generative AI for data type generation, also a Thai language translation to English.

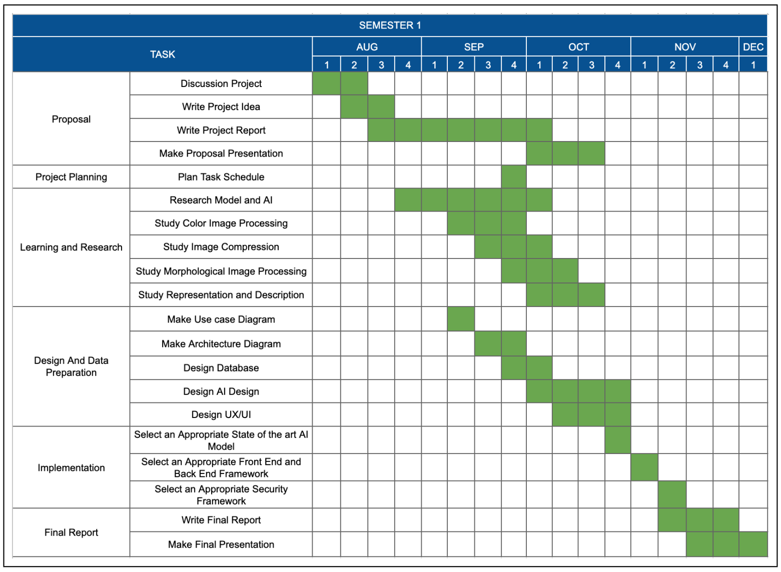
1.5 Limitation of Project

The limitation of the project will addresses a possible constraints and challenges that might affect its scope, execution, or outcomes. the limiting factor are include time, cost and risk etc.

- **OCR Accuracy:** The accuracy of OCR text extraction may vary depending on the quality of the uploaded document, such as low-resolution images, poor lighting, and the project does not support handwritten text.
- **Language Support:** While the system has the ability to translate text, the accuracy and availability of supported languages may be limited, with the project currently supporting Thai and English form only.
- **Required User Reviewing:** After the text extraction and layout detection, the system required a user to review and correct a input label that the system have process. The limitation required of user reviewing because of the lack of OCR accuracy and the system error.
- **Data Privacy and Security:** Despite the implementation of verification and security measures, there may still be vulnerabilities related to handling sensitive data, which are continuously checked and updated.

1.6 Project Schedule

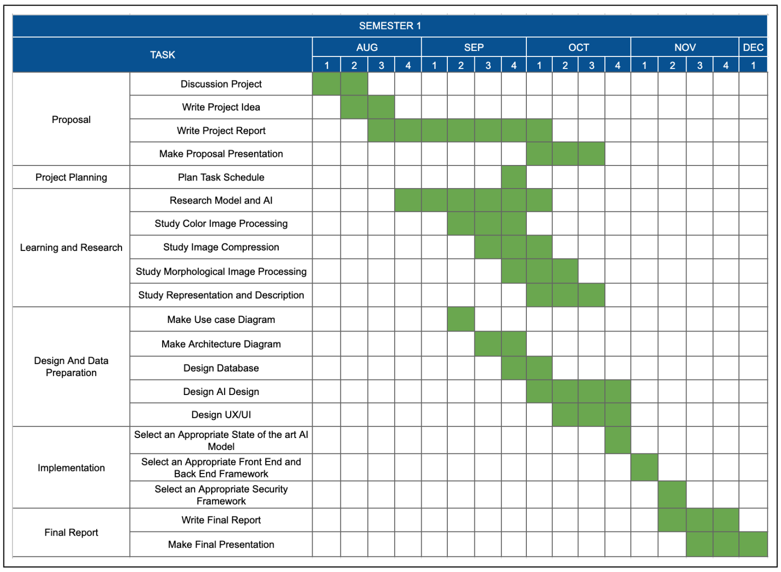
For the first semester, our project focus on researching and design phase, We have researching all core fundamental concept and define a problem and background of the project. In the design phase, we have design a database design, UX/UI design, architecture design. And this phase also including a Optical Character Recognition proof of concept.



SEMESTER 1																		
TASK		AUG				SEP				OCT				NOV				DEC
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1
Proposal	Discussion Project	🟩	🟩															
	Write Project Idea		🟩	🟩														
	Write Project Report			🟩	🟩	🟩	🟩	🟩	🟩	🟩								
	Make Proposal Presentation									🟩	🟩	🟩						
Project Planning	Plan Task Schedule								🟩									
Learning and Research	Research Model and AI				🟩	🟩	🟩	🟩	🟩	🟩								
	Study Color Image Processing						🟩	🟩	🟩									
	Study Image Compression							🟩	🟩	🟩								
	Study Morphological Image Processing								🟩	🟩	🟩							
	Study Representation and Description									🟩	🟩	🟩						
Design And Data Preparation	Make Use case Diagram						🟩											
	Make Architecture Diagram							🟩	🟩									
	Design Database								🟩	🟩								
	Design AI Design									🟩	🟩	🟩	🟩					
	Design UX/UI										🟩	🟩	🟩					
Implementation	Select an Appropriate State of the art AI Model												🟩					
	Select an Appropriate Front End and Back End Framework													🟩				
	Select an Appropriate Security Framework														🟩			
Final Report	Write Final Report														🟩	🟩	🟩	
	Make Final Presentation															🟩	🟩	🟩

Figure 1.1 Schedule for first semester

For the second semester, our project focus on implementing the form extractor and generative AI for the process of detect a form. We also focus on web application development and integrate a web application with the form extractor. also including the testing phase a system evaluation.



TASK		SEMESTER 1																
		AUG				SEP				OCT				NOV				DEC
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1
Proposal	Discussion Project	🟩	🟩															
	Write Project Idea		🟩	🟩														
	Write Project Report			🟩	🟩	🟩	🟩	🟩	🟩	🟩								
	Make Proposal Presentation									🟩	🟩	🟩						
Project Planning	Plan Task Schedule								🟩									
Learning and Research	Research Model and AI				🟩	🟩	🟩	🟩	🟩	🟩								
	Study Color Image Processing						🟩	🟩	🟩									
	Study Image Compression							🟩	🟩	🟩								
	Study Morphological Image Processing								🟩	🟩	🟩							
	Study Representation and Description									🟩	🟩	🟩						
Design And Data Preparation	Make Use case Diagram						🟩											
	Make Architecture Diagram							🟩	🟩									
	Design Database								🟩	🟩								
	Design AI Design									🟩	🟩	🟩	🟩					
	Design UX/UI										🟩	🟩	🟩					
Implementation	Select an Appropriate State of the art AI Model												🟩					
	Select an Appropriate Front End and Back End Framework													🟩				
	Select an Appropriate Security Framework														🟩			
Final Report	Write Final Report														🟩	🟩	🟩	
	Make Final Presentation															🟩	🟩	🟩

Figure 1.2 Schedule for second semester

1.7 Expected Outcomes

This project aims to develop a fully functional web application that able to converting a paper-based form or pdf form into a web-based form by utilizing a generative AI for generate a data type of form label. And the web application should reduce a time developer have spend when they converting a form.

CHAPTER 2 BACKGROUND THEORY AND RELATED RESEARCH

2.1 Introduction and Background

2.1.1 Introduction

This chapter will explain the details of the core concept and the solution planning. Theory and core concepts, languages and technologies, and related research will be discussed in this chapter. First, we will cover a core concept of artificial intelligence, machine learning, natural language processing and image processing etc.. Second, the languages and technologies that we interest in the project including a frontend and backend technology. Lastly, related research and competing solutions that are similar to our project will be in research and competing solutions.

2.1.2 Background

The digital transformation of businesses has been accelerated by the need for faster, more reliable ways to handle data. Although many organizations have begun to adopt digital processes, a significant number still rely on paper forms, which can slow down operations and increase the risk of errors. Manual data entry, in particular, is an inefficient method that often leads to mistakes, misinterpretations, and lost time.

To solve these problems, many organizations are looking for ways to turn paper forms into digital formats automatically. This is where AI comes in. AI tools can be trained to read forms and convert them into digital versions, speeding up the process and reducing errors. By automating this task, businesses can save time, lower costs, and reduce mistakes, allowing employees to focus on more important tasks.

This chapter will explain the main ideas behind the project. It will also discuss the technologies used in the project and look at similar research in the field of form automation.

2.2 Theory and Core Concepts

2.2.1 Artificial Intelligence (AI)

Artificial Intelligence (AI) refers to the study and development of intelligent machines and software that can reason, learn, communicate, and perceive objects, aiming to mimic human-like behavior. Coined by John McCarthy in 1956, AI is a branch of computer science that focuses on enabling computers to perform tasks typically requiring human intelligence, such as problem-solving, perception, and decision-making.

2.2.2 Machine Learning (ML)

Machine Learning is an application of Artificial Intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine Learning is crucial in building systems that can automatically learn to recognize patterns and improve over time with more data.

Machine Learning Method:

- **Supervised Learning:** A type of machine learning where the model is trained using a labeled dataset. This means the data comes with answers, so the model learns to make predictions or classify information correctly.

- **Unsupervised Learning:** A type of machine learning where the computer learns from data without labels or answers. Instead of being told what to look for, the model tries to find patterns or group similar data points together on its own.

- **Semi-supervised learning:** Like a mix of supervised and unsupervised learning. It uses a small amount of labeled data (with answers) and a large amount of unlabeled data (without answers) to train the model. The labeled data helps guide the model, while the unlabeled data helps it find patterns and improve accuracy.

2.2.3 Computer Vision (CV)

Computer Vision (CV) is a field of AI that uses machine learning to enable computers to interpret and understand visual information from the world, such as images and videos. It combines different methods and technologies.

Key functions of computer vision include analyzing images and videos to extract important information, understanding events and descriptions, and identifying patterns in scenery. CV employs methods that handle large volumes of data, making it applicable across various domains.

2.2.4 Image Processing

Image processing is the one technique in **Computer Vision (CV)** that is used to enhance and prepare images for analysis by applying various computational algorithms. In the context of form conversion, image processing plays a critical role in improving the quality of scanned documents or digital images before they undergo text recognition.

2.2.4.1 Image Acquisition

The first step in image processing is acquiring the image, where an image is captured using a camera, scanner, or another device. Then the image is converted into a digital format that can be manipulated by algorithms.

2.2.4.2 Preprocessing

Preprocessing is a fundamental step where the raw image quality is improved to ensure it is ready for more advanced analysis. This involves techniques such as:

- **Image Enhancement:** Image enhancement involves using techniques to improve the visual quality of an image so that important features are more visible. This is especially important when working with scanned documents or photos that may have different lighting conditions, colors, or other visual characteristics that could obscure the text. Here are some common techniques used for image enhancement:

  1. **Grayscale Conversion**
  2. **Histogram Equalization**
  3. **Thresholding**
  4. **Contrast Adjustment**

- **Image Restoration:** The Image Restoration process aims to improve the appearance of an image by fixing problems like noise and blurriness. It uses mathematical models to understand how the image got damaged and applies techniques to clean it up. Common techniques involved in image restoration include Noise Reduction and Blur removal

2.2.4.3 Segmentation

The main goal of image segmentation is to divide an image into distinct parts that match real objects or areas in the picture. This process separates important pixels from the rest of the image, resulting in sections that cover the entire image or outlines of objects.

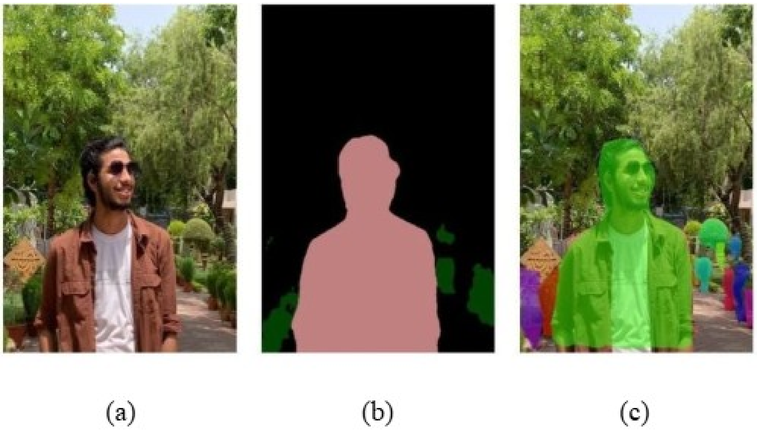


Figure 2.1 Image Segmentation Explained

(a) Original image

(b) Semantic Segmentation

(c) Instance Segmentation

2.2.4.4 Feature Extraction

Once the image has been segmented, the next step is to extract features that can be used for further analysis, such as character recognition or form field identification. Feature extraction techniques identify important attributes such as:

1. **Color Image Processing:** Color image processing is classified into two types:
   (a) **Full-color processing:**involves images captured using full-color sensors, further divided into:
      i. Individual Component Processing: Each color component (like RGB) is processed separately before creating a composite image.
      ii. Direct Color Pixel Manipulation: Color pixels are manipulated directly.

   (b) **Pseudo-color processing:** assigns colors to specific gray values based on criteria, utilizing techniques like intensity slicing and color coding.

2. **Image Compression:** Image compression reduces the amount of information needed to represent a digital image, primarily to save storage space or reduce bandwidth during transmission. It is categorized into two types:
   (a) **Lossless Compression:** No information is lost during compression.

   (b) **Lossy Compression:** Accepts some loss of information for higher compression rates.

3. **Morphological Image Processing:** This technique focuses on extracting and analyzing the shape and structure of objects within an image. Common morphological operators include:

   (a) Boundary extraction
   (b) Region filling
   (c) Skeletonization
   (d) Extraction of connected components

4. **Representation and Description:** Post-segmentation, raw pixel data needs to be compacted for further analysis. Representation techniques can focus on external features (like boundaries) or internal features (like pixels covering the region). Common methods include:
   (a) **Chain Codes and Polygonal Approximations:** representing shapes.
   (b) **Boundary Descriptors:** Include features like length, diameter, and curvature.
   (c) **Regional Descriptors:** Describe image regions in terms of area, compactness, gray level statistics, and topological features.

2.2.4.5 Object Recognition

Object recognition is the process of identifying and classifying different regions in an image, referred to as patterns or objects. There are two main approaches to object recognition

1. **Decision-Theoretic Approach:** This method uses numerical descriptions to analyze patterns. It looks at measurable features.

2. **Structural Approach:** This approach focuses on qualitative descriptions, using relational descriptors that describe the relationships and arrangements between different parts of the patterns.

2.2.5 Optical Character Recognition (OCR)

OCR or Optical Character Recognition is an input device used to read a printed text. OCR scans text in image, analyzes it character by character and converts it into machine-readable code and stores text on the system memory and processes it as a document. OCR is widely used in applications where transforming printed material into editable text.

2.2.5.1 OCR Architecture

After **image processing**, OCR comes into play. It extracts the text from the processed images, analyzing the document character by character to convert it into machine-readable code. It converts printed or handwritten text from forms into digital text, which can then be manipulated and stored. Once OCR extracts the raw text, **text processing** helps clean, format, and analyze this text.

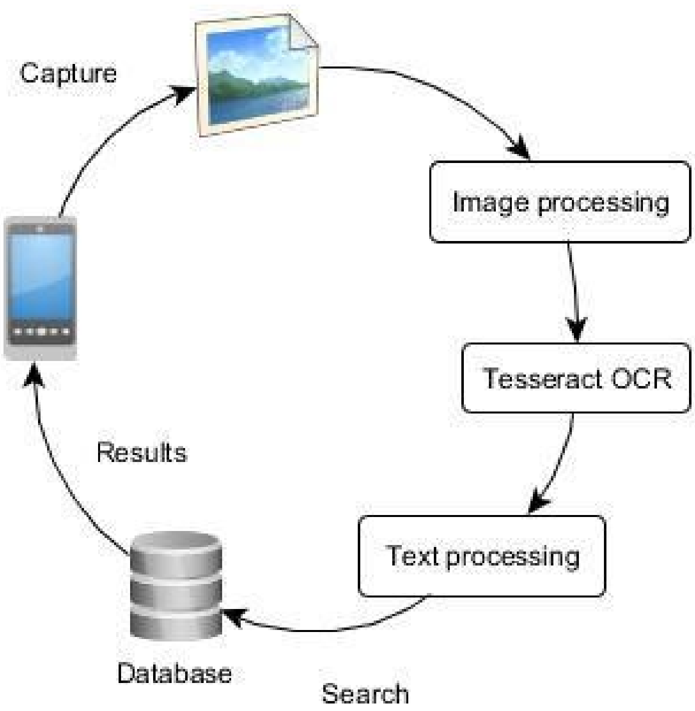


Figure 2.2 OCR Process

2.2.6 Natural Language Processing (NLP)

Natural Language Processing (NLP) enables machines to understand, interpret, and generate human language in a valuable way. The process involves several key techniques: Tokenization: Breaking down text into smaller units (tokens), such as words or phrases, which can be analyzed separately. Stop Word Removal: Eliminates common words that provide little informational value, focusing on more meaningful words. Lemmatization and Stemming: Reduces words to their root forms to unify different inflected versions (e.g., ”walking” becomes ”walk”). Part-of-Speech Tagging: Assigns tags to words based on their grammatical roles (nouns, verbs, adjectives, etc.).

2.2.7 Deep Learning

Deep Learning is a type of machine learning where computers are trained to think and learn like humans by processing large amounts of data. It uses a special kind of program called a neural network, which is inspired by the way our brains work. These networks can figure out patterns and make decisions on their own, such as recognizing faces in photos, translating languages, or even predicting what you might want to buy.

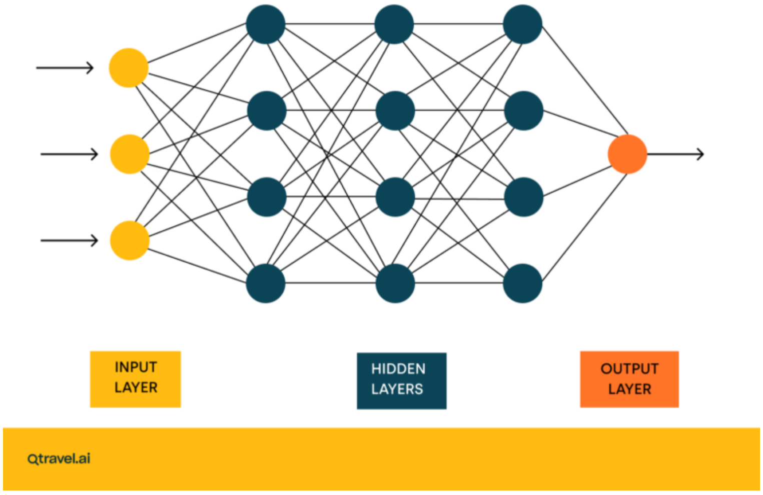


Figure 2.3 Natural Network

2.2.8 Generative AI

Generative AI refers to deep learning models that can generate high-quality text, images, and other content based on trained data. It works by analyzing a lot of data, such as Wikipedia articles or famous paintings, and then using what it learns to produce original outputs similar to the data it studied.

In the past, AI models like Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) helped create realistic images and sounds. Now, newer models like transformers have taken over. Transformers are powerful because they can learn from massive amounts of text without needing people to label everything.

2.2.8.1 Types of Transformer models

1. **Encoder-only models** (such as BERT): Great for understanding and categorizing text but not for generating new content.

2. **Decoder-only models** (such as GPT): Focus on predicting the next word and are good at creating new text.

3. **Encoder-decoder models** (such as T5): Combine both approaches and can handle tasks like summarizing or translating text.

2.2.9 Security of Web Form

Web form security is essential as these forms are highly vulnerable to cyberattacks, making them prime targets for malicious parties aiming to access sensitive user data. Common threats include:

1. **Cross-Site Scripting (XSS):** Attackers inject malicious scripts into web forms or application code, which can steal user information like keystrokes or cookies, or redirect users to harmful websites.

2. **SQL Injection:** Attackers manipulate SQL code within web forms to access or alter databases. This technique can expose or modify sensitive data, and it's one of the oldest and most dangerous web vulnerabilities.

3. **Cross-Site Request Forgery (CSRF):** Attackers trick users into performing unintended actions, such as submitting forms or transferring data, without their knowledge. These attacks exploit the user's browser and session data.

4. **Data Breaches:** A variety of attacks can lead to breaches, exposing sensitive customer information and damaging a business's reputation, as users lose trust in the website's security.

Many data privacy laws now regulate how web applications must ensure security, especially when handling personal information. The rules we must follow may depend on where our business is located and how it operates. To stay compliant, website owners need to adopt secure practices for web forms.

Key laws to be aware of

1. **GDPR (General Data Protection Regulation)**

2. **CCPA, CPRA (California Consumer Privacy Act and California Privacy Rights Act)**

3. **HIPAA (Health Insurance Portability and Accountability Act)**

4. **PCI DSS (Payment Card Industry Data Security Standard)**

Key measures for securing web forms

1. **Use TLS/SSL Certificates:** TLS/SSL certificates create encrypted links between browsers and servers, ensuring that data transferred is protected. Extended Validation (EV) certificates provide the highest level of security.

2. **Encrypt Data:** End-to-end encryption (E2EE) ensures that data is protected throughout its journey from sender to recipient, preventing unauthorized access. Use SSL certificates and CDNs to enhance security.

3. **Validate and Sanitize Input:** Validate and sanitize user inputs to prevent attacks like SQL injections or cross-site scripting (XSS). Validation checks if inputs are correct, while sanitization removes harmful data.

4. **Collect Minimal Data:** Only collect necessary information to reduce the impact of potential data breaches. This minimizes risk and aligns with privacy regulations like GDPR and CCPA.

5. **Anonymize Data:** Mask sensitive data (e.g., credit card numbers) using techniques like asterisks or data shuffling, making it harder for attackers to misuse it.

6. **Ask for Consent:** Obtain clear user consent before collecting personal data, and provide easy access to your privacy policy outlining how data is used and protected.

7. **Restrict File Uploads:** Limit file uploads by allowing only authorized users, setting file size limits, and using whitelist file types to prevent malware.

8. **Use reCAPTCHA:** Implement reCAPTCHA to differentiate between humans and bots, helping reduce spam and prevent automated attacks.

9. **Require Authentication:** For sensitive forms, require users to authenticate (e.g., through login or multi-factor authentication) to restrict access to authorized individuals only.

10. **Use Virus and Malware Protection:** Employ virus and malware protection through layered security, such as firewalls, malware scans, and intrusion detection, to safeguard your web forms from threats.

2.3 Languages and technologies

2.3.1 Web Development Language

2.3.1.1 HTML

HTML (HyperText Markup Language) is the standard language for creating and designing web pages. It is a markup language, which means that specific tags are used to structure web content. Each tag specifies a particular aspect of a website, such as text, photos, links, and multimedia.

2.3.1.2 TypeScript

TypeScript is a superset of JavaScript that introduces static typing. It was created by Microsoft to solve some of the limitations of JavaScript, particularly in large-scale applications. Because TypeScript is a superset, it supports all JavaScript features while also adding new functionalities, notably those aimed at increasing code dependability and maintenance.

2.3.2 Front-end Technology

2.3.2.1 Vue.js

Vue.js is a JavaScript framework for building user interfaces. It builds on top of standard HTML, CSS, and JavaScript and provides a declarative, component-based programming model that helps you efficiently develop user interfaces of any complexity.

2.3.2.2 TailwindCSS

Tailwind CSS is a utility-focused CSS framework that simplifies web development by providing pre-designed utility classes. These utility classes allow you to create custom designs.

2.3.2.3 SurveyJS

SurveyJS is a set of JavaScript components that allow users to build surveys, quizzes, polls, and other web forms, store them in your database, and visualize survey results in custom dashboards.

2.3.3 Backend Technology

2.3.3.1 FastAPI

FastAPI is a modern web framework for creating Python APIs that prioritizes speed, ease of development, and automated validation. It has gained popularity for developing quick, high-performance APIs because it uses Python type hints to provide explicit, automatic data validation and interactive API documentation.

2.3.3.2 Pydantic

Pydantic is a Python data validation and parsing library that enforces data types and restrictions using Python type annotations. Although it can be used separately, it is an essential part of FastAPI. With Pydantic models, you can easily serialize data, validate incoming data, and specify the structure of your data.

2.3.3.3 SQLAlchemy

SQLAlchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL without creating a SQL statement.

2.3.3.4 PostgresSQL

The PostgreSQL popular open-source relational database management system (RDBMS) for managing and storing data. It is renowned for being reliable, effective, and compliant with SQL standards. And It support storing a json format data type which it can be act as NoSQL in some situation.

2.3.4 Document Processing

2.3.4.1 Python

Python Programming language is a popular high-level computer programming language for machine learning because it is simple to use and easy to read and write. Additionally, it has a fast processing speed. Web development, data analysis, automation, scientific computing, and many more topics are among the many libraries available.

2.3.4.2 Open Source Computer Vision Library (OpenCV)

Open Source Computer Vision Library is an open-source library designed for real-time computer vision and image processing. It provides tools for tasks such as object detection, image segmentation, facial recognition, and edge detection. OpenCV is widely used in fields like robotics, augmented reality, and machine learning.

OpenCV processes image data using built-in functions for manipulating pixel values. It works with image and video formats and can be integrated with machine learning models to perform visual recognition tasks. It supports various languages, including Python

2.3.4.3 Tesseract OCR

Tesseract OCR is an open-source Optical Character Recognition (OCR) engine used to convert images of text into machine-encoded text. Developed by Hewlett-Packard and maintained by Google, it supports various languages and one of them is python, making it highly versatile for tasks like digitizing printed materials, extracting text from images, and enhancing accessibility in digital applications. It is commonly integrated with other image processing libraries like OpenCV to handle pre-processing tasks, improving text recognition accuracy, especially in noisy or distorted images.

2.3.4.4 NLLB-200

NLLB-200 (No Language Left Behind) is a project by Meta (Facebook) that focuses on creating AI models capable of translating between 200 languages.

2.3.4.5 PyTorch

PyTorch is an open-source deep learning framework developed by Facebook's AI Research lab (FAIR). It provides tools for building and training neural networks using dynamic computational graphs, which allow for flexibility and ease in model building.

2.3.4.6 LLAMA

LLama (short for Large Language Model Architecture) is a type of AI model designed to understand and generate human-like text. It is built using a transformer architecture, which helps it learn language patterns and produce meaningful responses based on its training data.

2.4 Competing solutions

2.4.1 Microsoft Azure Document Intelligence

Figure 2.4 Microsoft Azure Document Intelligence

Microsoft Azure Document Intelligence is a solution that provides a feature of capturing data from printed or handwritten forms. And create a flow pipeline with Azure Cognitive Search pipeline to complete workflow as the user needs.

CHAPTER 3 DESIGN AND METHODOLOGY

This chapter will cover the features, architecture, functionalities, design methods, and diagrams of our web application. We will delve into the details of the application's functionality and architecture.

3.1 Project Functionality

3.1.1 System Requirements

- The web application must allow users to log in with email and password.
- The web application must allow the user to upload a form file to the system.
- The web application must allow users to fill the form without login.
- The web application must allow users to view the data of the form.
- The web application must allow users to edit the form before publishing.
- The web application must allow users to delete the form.
- The web applications must provide the option for all logged-in users to logout.

3.1.2 Feature List

3.1.2.1 Paper-Based Form Analysis System

The Paper-Based Form Analysis System will extract all the text from the document that is uploaded by the user via a web application. The system will then analyze the form's pattern ,extract the input labels, translate the text into English, and send the translated information to a generative AI for data type generation.

3.1.2.2 Form Schema generator System

The Form Schema generator System will receive information from the Paper-Based Form Analysis System and Form Schema generator System must be able to generate a form schema that compatible with a form library.

3.1.2.3 Web Form System

The Digital Form System must enable users to complete the form, store the data in the database, and see the data created by the form owner.

3.1.2.4 Registration and Authentication System

The Registration and Authentication System must enable a user to register and login to the system by using only email and password. This including a forgot password and the OTP for reset password.

3.2 Use Case Diagram

From the figure 3.1 The diagram shows a relationship between the user and the system by using a use case diagram. The user of the system is a developer that needs to transform a paper-base form into web application form and the user who going to fill the form. The system consists of 3 different systems, which are paper-based form analysis systems, form schema generator systems, web form systems and registration and authentication systems. The user can upload a paper-based form to the system, and the system will transform the form into a web-based form.

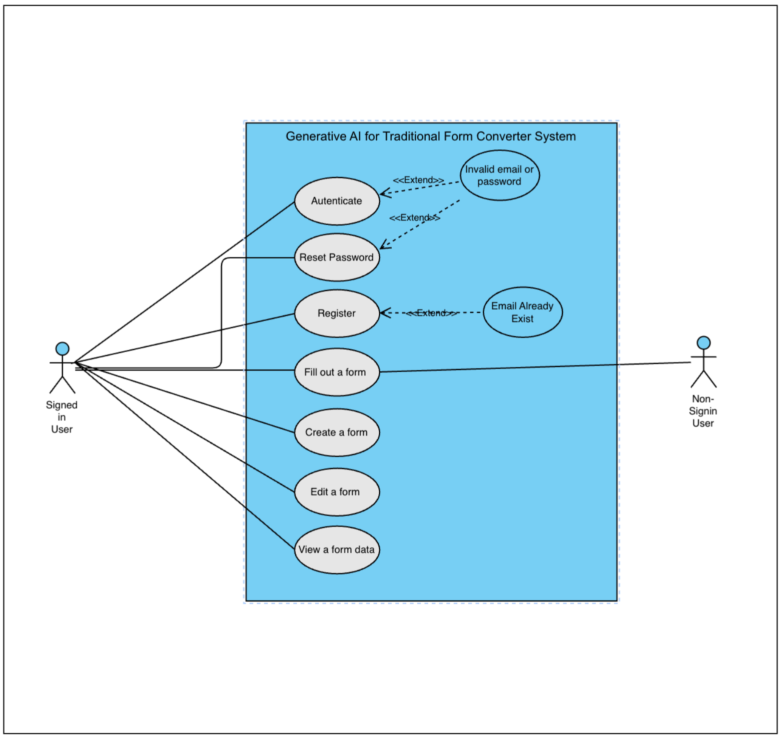


Figure 3.1 Use case diagram

3.3 Use Case Narrative

3.3.1 Autentication

Use Case Name: Autentication

Actors: Form Creator and Required Login User

Goal: Users log in to the system.

Preconditions: User is registered

Main Success Scenario:

1. User access the website.
2. User enter a email and password.
3. User submit a email and password.
4. System authenticate and navigate to the home page.

3.3.2 Register

Use Case Name: Register

Actors: Form Creator and Required Login User

Goal: Users register to a system.

Preconditions: User does not have an account

Main Success Scenario:

1. User access the website.
2. User click to create a new account.
3. User enter an email and password and personal information.

4. User submit information.
5. System saves user information and navigates users to the login page.

3.3.3 Create a form

Use Case Name: Create a form

Actors: Form Creator

Goal: Create a form by upload the form file.

Preconditions: User has logged in.

Main Success Scenario:

1. User go to home page.
2. User click at upload a form.
3. User select a file to upload.
4. System will process the file and navigate users to the edit form page to confirm a form before publishing.

3.3.4 Edit a form

Use Case Name: Edit a form

Actors: Form Creator

Goal: Edit a form to make a change.

Preconditions: User has logged in.

Main Success Scenario:

1. User go to home page.
2. User click at edit a form at the form user need to make change.
3. User make change a form.
4. User click back to the previous page.
5. System will process autosave and navigate users to the previous page.

3.3.5 View a form data

Use Case Name: View a form data

Actors: Form Creator

Goal: View the data that user have input

Preconditions: User has logged in.

Main Success Scenario:

1. User go to home page.
2. User click at view a data of the form.
3. User can see a form data

3.3.6 Fill up the form

Use Case Name: Fill up the form

Actors: Required Login Users and Anonymous User

Goal: Add a new data to the form

Preconditions: User has logged in or non-login user.

Main Success Scenario:

1. User access a form via the public link
2. User fill up a form.
3. User submit a form data.
4. System will save the data and navigate to the form page again.

Alternate scenario (user access the form required a login without login):

1. User access a form via the public link
2. System will navigate to the login page After login completes the user will redirect back to the form page.

3.4 Activity Diagram

From Figure 3.2 The Activity diagram shows the sequence how Generative AI for Traditional Form Converter System is working.

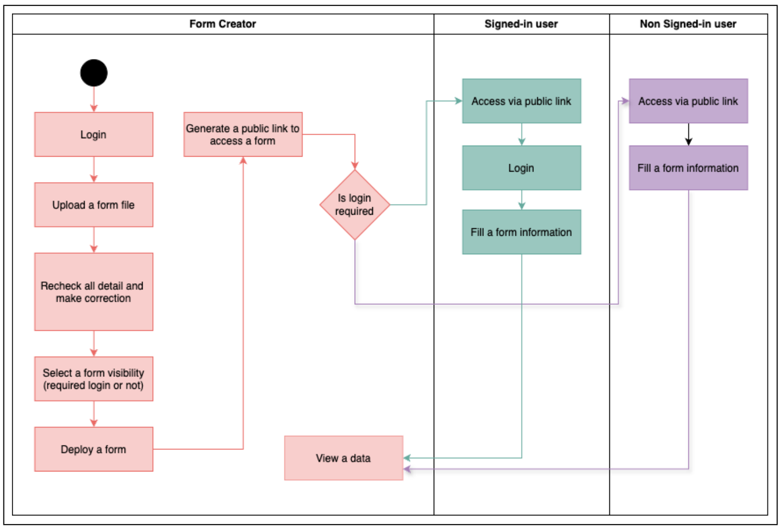


Figure 3.2 Activity Diagram

3.4.1 Form creator

- **Login:** Form Creator must login before use the system
- **Upload a form file:** After logging in, Form creator must upload a form file to add new form to the system.

- **Recheck all detail and make correction:** In this step, Form creator must check all the information that system has generated and make a correction if incase of error text found.
- **Select a from visibility:** Select a form visibility, whether the form creator need to form to be access by the user who signed-in or anyone can access.
- **Deploy a form and Generate a link:** In this step, the form will be saved and generated a link to allow user to access.

3.4.2 Signed-in user

- **Login:** If the form requires login, the user must log into the system.
- **Fill out information:** Once logged in, the user going to fills out the form and submit.

3.4.3 Non Signed-in user

- **Fill out information:** If login is not required, the user can directly fill out the form without logging in.

3.5 System Architecture

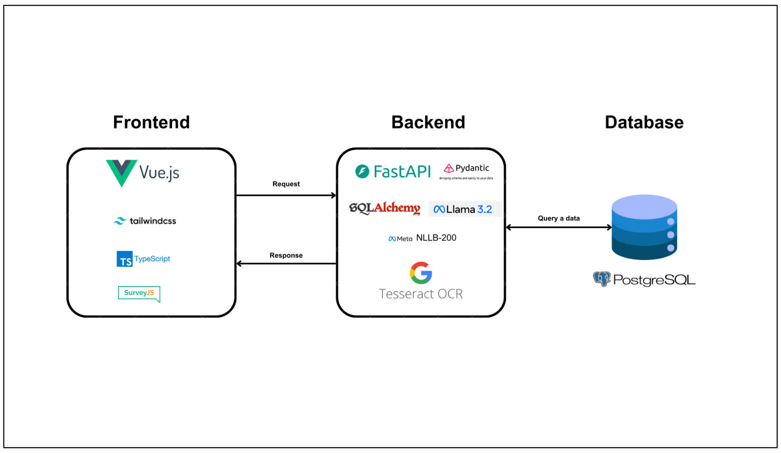


Figure 3.3 System Architecture

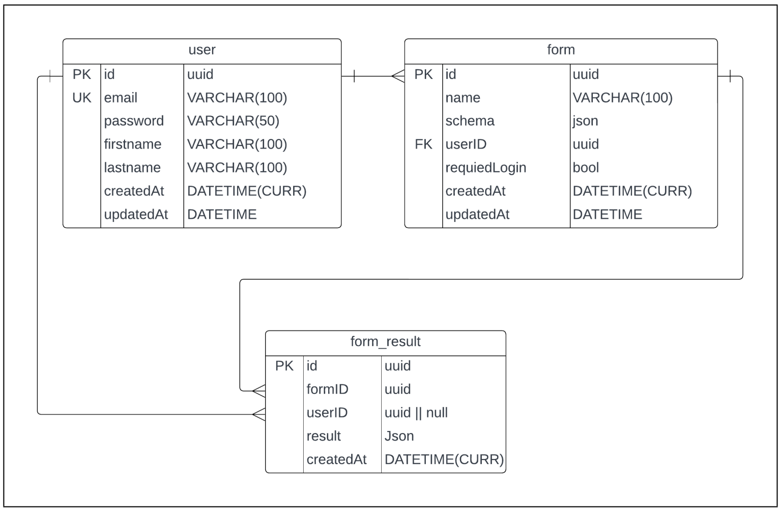
Figure 3.3, The diagram shown a system architecture in figure above, The System have divided into 3 part which front-end, back-end and database, each part have shown a technology stack, and here are the description of each component

- **VueJS** is a Front-end JavaScript library for building UI
- **TailwindCSS** is a CSS framework for styling the UI and used with React
- **FastAPI** is a Back-end framework for building REST API
- **PostgreSQL** is a relational database
- **Pydantic** is a python library used for data validation
- **SQLAlchemy** is a Python base Object Relational Mapper (ORM) and SQL Tool kit
- **SurveyJS** is a form engine Llibrary

- **Meta NLLB-200** is a Model for text translation
- **Tesseract OCR** is a OCR for extract text from image
- **Llama 3.2** is a generative AI from Facebook

3.6 Database Design

Figure 3.4 shown a project database design, the system consist three tables in our project database design are user, form, and formresult. table is used to store user data, formresult is used to store a result that the user has filled out, and form table is used to store a form schema.



user		
PK	id	uuid
UK	email	VARCHAR(100)
	password	VARCHAR(50)
	firstname	VARCHAR(100)
	lastname	VARCHAR(100)
	createdAt	DATETIME(CURR)
	updatedAt	DATETIME

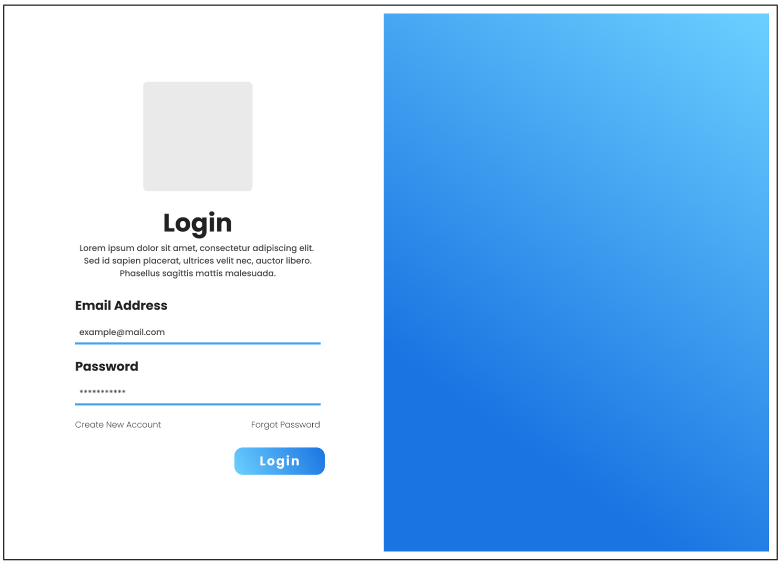
form		
PK	id	uuid
	name	VARCHAR(100)
	schema	json
FK	userID	uuid
	requiedLogin	bool
	createdAt	DATETIME(CURR)
	updatedAt	DATETIME

form_result		
PK	id	uuid
	formID	uuid
	userID	uuid || null
	result	Json
	createdAt	DATETIME(CURR)

Figure 3.4 ER Diagram

3.7 User Interface Design

3.7.1 Login Page



Login

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed id sapien placerat, ultrices velit nec, auctor libero. Phasellus sagittis mattis malesuada.

Email Address

example@mail.com

Password

****

Figure 3.5 Login Page

Figure 3.5 represents the login screen of the web application. This page have a email field, password field and a login button to sent a credential to the back-end system.

3.7.2 Create Your Account

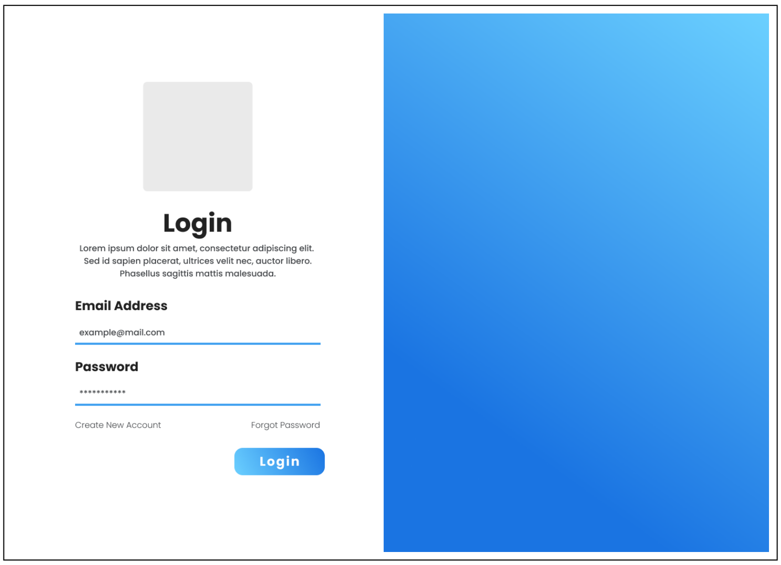

Login
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed id sapien placerat, ultrices velit nec, auctor libero. Phasellus sagittis mattis malesuada.
Email Address
example@mail.com
Password
•••••••••••
Create New Account
Forgot Password
Login

Figure 3.6 Create Account Page

Figure 3.6 represents the create account page of the web application. This page allow user to create their own account by the user must provide a following field which is name, email and password.

3.7.3 Forgot Password

At Figure 3.7 represents the create account page of the web application. When the user forgot their password, The user must navigate to this page by click at forget password from login page. And fill the email address to allow the system sent the One-time password (OTP) to email address.

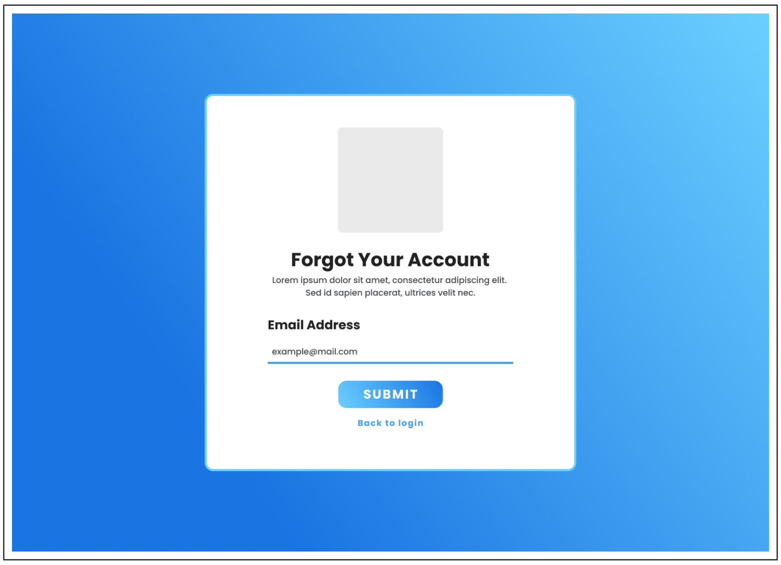
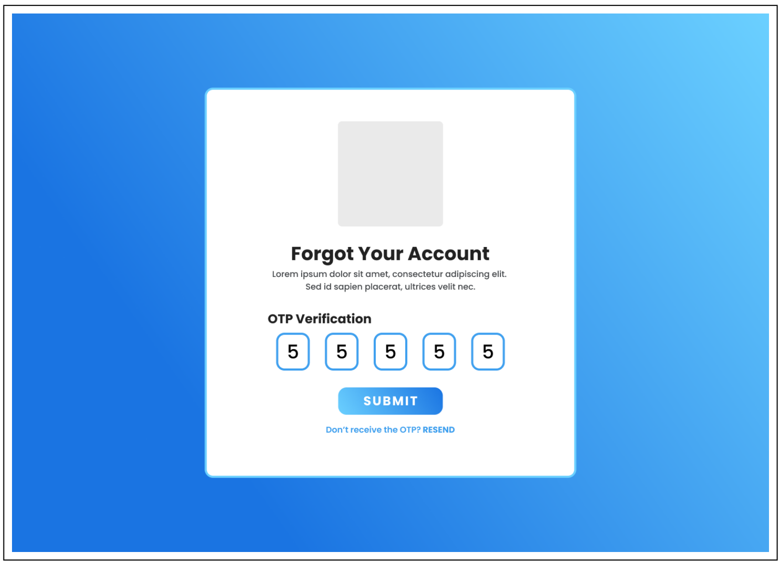

Forgot Your Account
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed id sapien placerat, ultrices velit nec.
Email Address
example@mail.com
SUBMIT
Back to login

Figure 3.7 Forgot Password Page



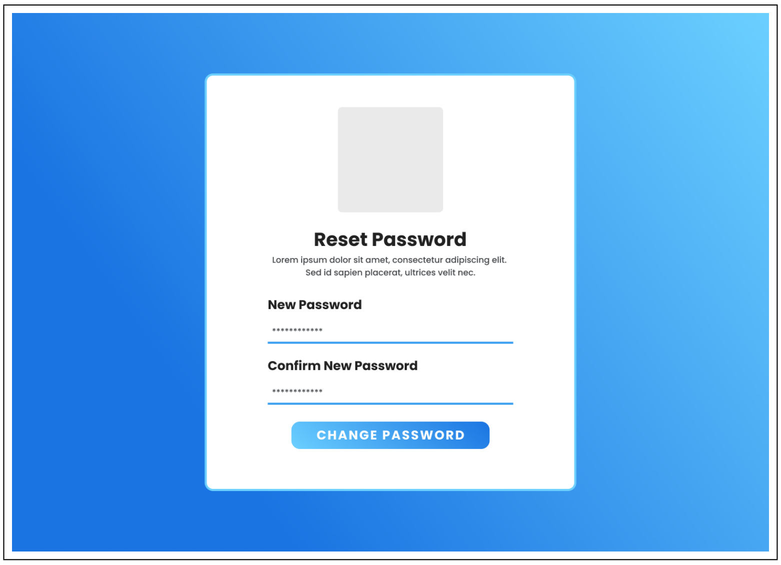
Forgot Your Account

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed id sapien placerat, ultrices velit nec.

OTP Verification

5 5 5 5 5

Figure 3.8 Forgot Password Page



Reset Password

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed id sapien placerat, ultrices velit nec.

New Password

••••••••••••

Confirm New Password

••••••••••••

Figure 3.9 Reset Password Page

At the Figure 3.8, It represents a OTP confirmation page which required a OTP Code that the system have send to the email address. Figure 3.9, It represents a reset password pageIf it success the system will navigate to reset password page for enter a new password.

3.7.4 Dashboard

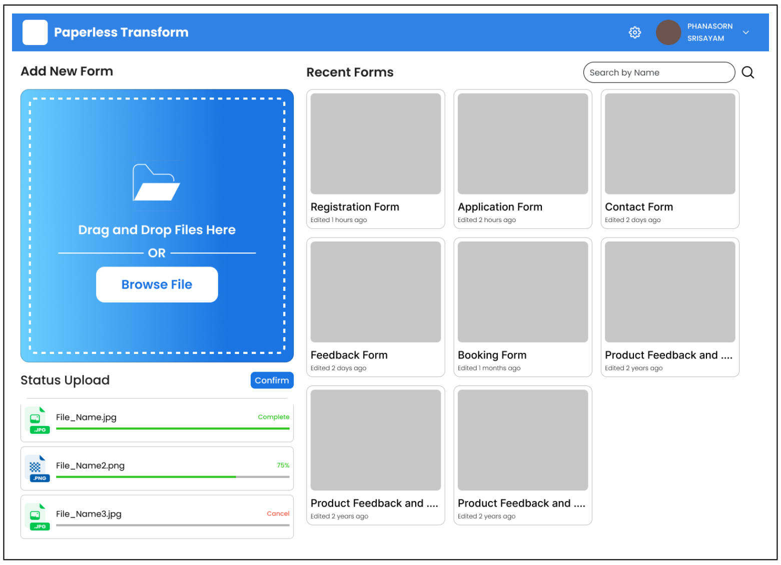
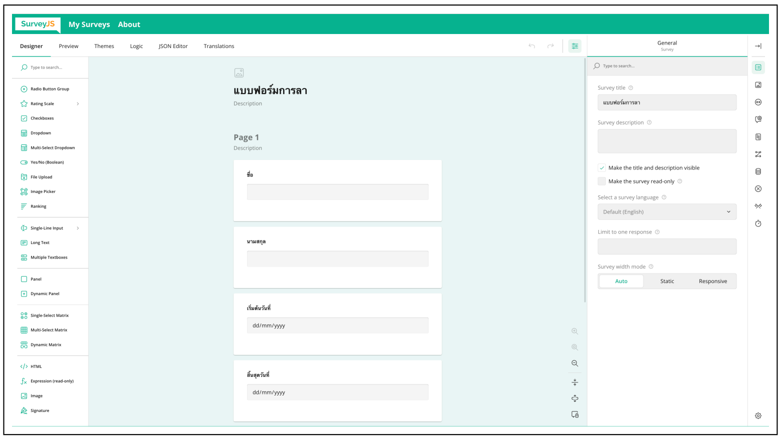


Figure 3.10 Dashboard Page

Figure 3.10 represents the dashboard page, it will show all the form that user have and the add form section at the left hand side, also have a upload status while file is uploading.

3.7.5 Edit Form



แบบฟอร์มการลา

Description

Page 1

Description

ชื่อ

นามสกุล

เริ่มต้นวันที่

dd/mm/yyyy

สิ้นสุดวันที่

dd/mm/yyyy

Survey title: แบบฟอร์มการลา

Survey description

☑ Make the title and description visible

☐ Make the survey read-only

Select a survey language: Default (English)

Limit to one response

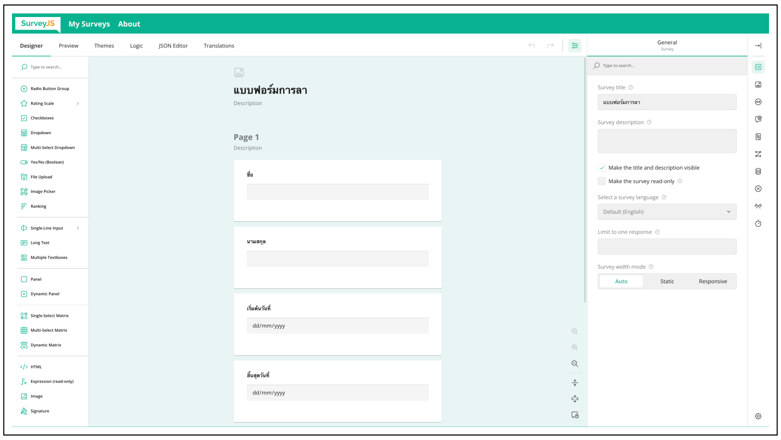
Survey width mode: Auto | Static | Responsive

Figure 3.11 Edit Form Page

Figure 3.10 represents the Edit Form. The edit form page uses the SurveyJS library, which allows users to fully customize a form, including changing the color, adding a second page, and creating form conditions.

3.7.6 Form

Figure 3.10 represents the Edit Form. The edit form page uses the SurveyJS library, which allows users to fully customize a form, including changing the color, adding a second page, and creating form conditions.



แบบฟอร์มการลา

Description

Page 1

Description

ชื่อ

นามสกุล

เริ่มต้นวันที่

dd/mm/yyyy

สิ้นสุดวันที่

dd/mm/yyyy

Survey title: แบบฟอร์มการลา

Survey description

☑ Make the title and description visible

□ Make the survey read-only

Select a survey language: Default (English)

Limit to one response

Survey width mode: [✓ Auto] Static Responsive

Figure 3.12 Edit Form Page

CHAPTER 4 IMPLEMENTATION RESULTS

You can title this chapter as **Preliminary Results** or **Work Progress** for the progress reports. Present implementation or experimental results here and discuss them.

ALL SECTIONS IN THIS CHAPTER ARE OPTIONAL. PLEASE CONSULT YOU ADVISOR AND DESIGN YOUR OWN SECTION

หัวข้อต่าง ๆ ในแต่ละบทเป็นเพียงตัวอย่างเท่านั้น หัวข้อที่จะใส่ในแต่ละบทขึ้นอยู่กับโปรเจคของนักศึกษาและอาจารย์ที่ปรึกษา

CHAPTER 5 CONCLUSIONS

Figure 5.1 This is how you mention when figure come from internet https://www.google.com

This chapter is optional for proposal and progress reports but is required for the final report.

THIS IS AN EXAMPLE. ALL SECTIONS BELOW ARE OPTIONAL. PLEASE CONSULT YOU ADVISOR AND DESIGN YOUR OWN SECTION

หัวข้อต่าง ๆ ในแต่ละบทเป็นเพียงตัวอย่างเท่านั้น หัวข้อที่จะใส่ในแต่ละบทขึ้นอยู่กับโปรเจคของนักศึกษาและอาจารย์ที่ปรึกษา

5.1 Problems and Solutions

State your problems and how you fixed them.

5.2 Future Works

What could be done in the future to make your projects better.

REFERENCES

1. I. Norros, 1995, “On the use of Fractional Brownian Motion in the Theory of Connectionless Networks,” **IEEE J. Select. Areas Commun.**, vol. 13, no. 6, pp. 953–962, Aug. 1995.

2. H.S. Kim and N.B. Shroff, 2001, “Loss Probability Calculations and Asymptotic Analysis for Finite Buffer Multiplexers,” **IEEE/ACM Trans. Networking**, vol. 9, no. 6, pp. 755–768, Dec. 2001.

3. D.Y. Eun and N.B. Shroff, 2001, “A Measurement-Analytic Framework for QoS Estimation Based on the Dominant Time Scale,” in **Proc. IEEE INFOCOM’01**, Anchorage, AK, Apr. 2001.

APPENDIX A

FIRST APPENDIX TITLE

Put appropriate topic here

This is where you put hardware circuit diagrams, detailed experimental data in tables or source codes, etc..

Figure A.1 This is the figure x11 https://www.google.com

This appendix describes two static allocation methods for fGn (or fBm) traffic. Here, λ and C are respectively the traffic arrival rate and the service rate per dimensionless time step. Their unit are converted to a physical time unit by multiplying the step size Δ. For a fBm self-similar traffic source, Norros [1] provides its EB as

$$C = \lambda + (\kappa(H)\sqrt{-2\ln\epsilon})^{1/H} a^{1/(2H)} x^{-(1-H)/H} \lambda^{1/(2H)} \tag{A.1}$$

where $\kappa(H) = H^H(1-H)^{(1-H)}$. Simplicity in the calculation is the attractive feature of (A.1).

The MVA technique developed in [2] so far provides the most accurate estimation of the loss probability compared to previous bandwidth allocation techniques according to simulation results. Consider a discrete-time queueing system with constant service rate C and input process λ_n with $\mathbb{E}\{\lambda_n\} = \lambda$ and $\mathrm{Var}\{\lambda_n\} = \sigma^2$. Define $X_n \equiv \sum_{k=1}^{n} \lambda_k - Cn$. The loss probability due to the MVA approach is given by

$$\varepsilon \approx \alpha e^{-m_x/2} \tag{A.2}$$

where

$$m_x = \min_{n\geq 0} \frac{((C-\lambda)n + B)^2}{\mathrm{Var}\{X_n\}} = \frac{((C-\lambda)n^* + B)^2}{\mathrm{Var}\{X_{n^*}\}} \tag{A.3}$$

and

$$\alpha = \frac{1}{\lambda\sqrt{2\pi\sigma^2}} \exp\left(\frac{(C-\lambda)^2}{2\sigma^2}\right) \int_C^\infty (r - C) \exp\left(\frac{(r-\lambda)^2}{2\sigma^2}\right) dr \tag{A.4}$$

For a given ε, we numerically solve for C that satisfies (A.2). Any search algorithm can be used to do the task. Here, the bisection method is used.

Next, we show how $\mathrm{Var}\{X_n\}$ can be determined. Let $C_\lambda(l)$ be the autocovariance function of λ_n. The MVA technique basically approximates the input process λ_n with a Gaussian process, which allows $\mathrm{Var}\{X_n\}$ to be represented by the autocovariance function. In particular, the variance of X_n can be expressed in terms of $C_\lambda(l)$ as

$$\mathrm{Var}\{X_n\} = nC_\lambda(0) + 2\sum_{l=1}^{n-1}(n-l)C_\lambda(l) \tag{A.5}$$

Therefore, $C_\lambda(l)$ must be known in the MVA technique, either by assuming specific traffic models or by off-line analysis in case of traces. In most practical situations, $C_\lambda(l)$ will not be known in advance, and an on-line measurement algorithm developed in [3] is required to jointly determine both n^* and m_x. For fGn traffic, $\mathrm{Var}\{X_n\}$ is equal to $\sigma^2 n^{2H}$, where $\sigma^2 = \mathrm{Var}\{\lambda_n\}$, and we can find the n^* that minimizes (A.3) directly. Although λ can be easily measured, it is not the case for σ^2 and H. Consequently, the MVA technique suffers from the need of prior knowledge traffic parameters.

APPENDIX B

SECOND APPENDIX TITLE

Put appropriate topic here

Figure B.1 This is the figure x11 https://www.google.com

Next, we show how $\mathrm{Var}\{X_n\}$ can be determined. Let $C_\lambda(l)$ be the autocovariance function of λ_n. The MVA technique basically approximates the input process λ_n with a Gaussian process, which allows $\mathrm{Var}\{X_n\}$ to be represented by the autocovariance function. In particular, the variance of X_n can be expressed in terms of $C_\lambda(l)$ as

$$\mathrm{Var}\{X_n\} = nC_\lambda(0) + 2\sum_{l=1}^{n-1}(n-l)C_\lambda(l) \tag{B.1}$$

Add more topic as you need

Therefore, $C_\lambda(l)$ must be known in the MVA technique, either by assuming specific traffic models or by off-line analysis in case of traces. In most practical situations, $C_\lambda(l)$ will not be known in advance, and an on-line measurement algorithm developed in [3] is required to jointly determine both n^* and m_x. For fGn traffic, $\mathrm{Var}\{X_n\}$ is equal to $\sigma^2 n^{2H}$, where $\sigma^2 = \mathrm{Var}\{\lambda_n\}$, and we can find the n^* that minimizes (A.3) directly. Although λ can be easily measured, it is not the case for σ^2 and H. Consequently, the MVA technique suffers from the need of prior knowledge traffic parameters.