**Industry Project Report**

**On**

**Swift Notes: AI-Powered Study Notes Generator**

|  |  |
| --- | --- |
| Developed By: - | Guided By:- |
| Dushyant K Vyas (22162172006) | Dr. Pritesh Andharia (Internal) |
|  | Mr. Nirav Rajgor (IBM)(External) |
|  |  |

**Submitted to**

**Faculty of Engineering and Technology**

**Institute of Computer Technology**

**Ganpat University**

****

**Year - 2025**



# CERTIFICATE

This is to certify that the **Industry** Project work entitled **“Swift Notes: AI-Powered Study Notes Generator”** by Dushyant K Vyas (Enrollment No.22162172006) of Ganpat University, towards the partial fulfillment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, carried out by them in the CSE(CS) Department at Ganpat University. The results/findings contained in this Project have not been submitted in part or full to any other University / Institute for award of any other Degree/Diploma.

Name & Signature of Internal Guide

Name & Signature of Head

**Place: ICT - GUNI**

### Date: 26th April, 2025

**ACKNOWLEDGEMENT**

The Industry Internship Project has been an invaluable experience, offering a chance for both learning and personal growth. I feel truly privileged to have had the support and guidance of many remarkable individuals throughout this journey. I extend my sincere gratitude to **Prof. Rohit Patel**, Principal of ICT, and **Prof. Dharmesh Darji**, Head of ICT, for granting us the opportunity to undertake this project. I am especially thankful to **Dr. Pritesh Andharia** and **Mr. Nirav Rajgor**, our internal and external mentors, for their insightful guidance during the development of our project *"****Swift Notes: AI-Powered Study Notes Generator****".* Despite their demanding academic schedules, they consistently found time to guide us and ensure we stayed on the right track. Their mentorship was instrumental to the progress and success of our work. We also acknowledge the constant support of the **Computer Science and Engineering Department**, whose regular monitoring and timely arrangements provided a smooth and encouraging environment for project development. We deeply appreciate all their contributions.

### Dushyant K Vyas (Enrollment No: 22162172006)

**ABSTRACT**

Swift Notes is an AI-powered study notes generator that automates the process of extracting text from PDF documents, summarizing key points, and organizing them into structured study materials. This tool is designed to assist students by reducing manual note-taking efforts and enhancing learning efficiency. Using Google Gemini’s multimodal models like Gemini 1.5 Pro or Gemini 2.0 Flash, it provides quick, concise, and exportable study notes for academic and research use.

**INDEX**

### Title PageNo

**CHAPTER 2:PROJECT SCOPE 03-04**

**CHAPTER1:INTRODUCTION 01-02**

**CHAPTER 3: SOFTWARE AND HARDWARE REQUIREMENT 05-06**

**CHAPTER 4: PROCESS MODEL 07-08**

[CHAPTER 5: PROJECT PLAN 09-10](#_TOC_250009)

* 1. [Timeline 10](#_TOC_250008)

[CHAPTER 6: IMPLEMENTATION DETAILS 11-](#_TOC_250006)12

* 1. [Flowchart of Implementation 12](#_TOC_250005)

**CHAPTER 7: CONCLUSION AND FUTURE WORK 13-15**

**CHAPTER8:REFERENCE 16-17**

**CHAPTER: 1 INTRODUCTION**

**CHAPTER 1 INTRODUCTION**

In the fast-paced world of academics, students often face the overwhelming task of **manually summarizing and organizing** large volumes of study materials. Whether preparing for exams, reviewing lecture notes, or analyzing research papers, the process of extracting relevant information and creating structured summaries is **time-consuming and mentally exhausting**. Traditional note-taking methods require hours of reading, comprehension, and manual summarization, leaving students with less time to focus on actual learning.

Students often struggle to manage the massive amount of content they need to study. For example, a student preparing for competitive exams may have to review hundreds of pages of textbooks and research articles. This results in a **significant time investment** and **inconsistent note quality**, making it difficult to retain and revise key concepts efficiently.

In addition, many students rely on **manual processes** to create study notes, which are prone to human error and inefficiency. With the increasing availability of digital resources, there is a growing need for a **smarter, automated** solution that can process large documents and extract **concise, well-structured** information quickly.

**Swift Notes** is an AI-powered study notes generator designed to automate and streamline the process of creating structured summaries from PDF documents. Swift Notes leverages **Google's Gemini 1.5 Pro**, a cutting-edge large language model (LLM) capable of generating high-quality, concise, and contextually rich summaries. The system extracts meaningful content from uploaded PDFs, intelligently segments the text, and then uses Gemini to produce bullet-point summaries that are accurate, diverse, and easy to read — all within a clean, user-friendly web interface.

**CHAPTER: 2 PROJECT SCOPE**

### CHAPTER 2 PROJECT SCOPE

The **Swift Notes** project is designed to address the growing need for automated study material generation by providing an **AI-powered** solution that converts lengthy academic PDFs into concise, well-structured study notes. This project focuses on reducing the time and effort students spend manually summarizing large documents, allowing them to focus on learning and comprehension.

The primary functionality of Swift Notes involves extracting text from PDF documents, processing the extracted content through AI-driven summarization, and delivering user-friendly study notes. Users can upload PDF files through an intuitive web-based interface built using Django. Once uploaded, the system extracts and cleans the text, ensuring that irrelevant content like page numbers, headers, and footers is removed. The cleaned text is then processed by transformer models to generate meaningful summaries, which are formatted into bullet points and section-wise outputs for clarity. This structured presentation allows students to quickly grasp complex topics without reading the entire document.

Swift Notes also provides export capabilities, allowing users to download the generated summaries in both PDF and plain text formats. This feature ensures that students can access their study materials offline or share them easily. The system is optimized to handle large and multi-page documents efficiently by splitting them into manageable sections before summarization.

While the current scope is focused on English-language documents, future iterations of Swift Notes **may** expand to support **multilingual** summarization. The system is not yet designed to handle image-based or scanned PDFs that require **Optical Character Recognition (OCR),** though future improvements could also integrate OCR functionality. Another potential enhancement is **multi-PDF** batch processing, enabling users to summarize multiple documents simultaneously. Furthermore, Swift Notes can be extended to include customization options for adjusting the output’s tone, length, and structure, providing a personalized experience for users.

Technically, the project is built using Python as the core programming language. The backend is developed using **Django**, while the summarization process is powered by **Google’s Generative AI**. **PyMuPDF** is used for PDF parsing and extraction, and the export feature is implemented using **ReportLab**. The system is designed for deployment on cloud platforms like Render or Heroku, allowing public access and scalability.

# CHAPTER: 3 SOFTWARE AND HARDWARE REQUIREMENTS

### CHAPTER 3 SOFTWARE AND HARDWARE REQUIREMENTS

**Minimum Hardware Requirements**

|  |  |
| --- | --- |
| **Processor** | 2.0 GHz |
| **RAM** | 4GB |
| **HDD** | 4GB |

*Table 3.1 Minimum Hardware Requirements*

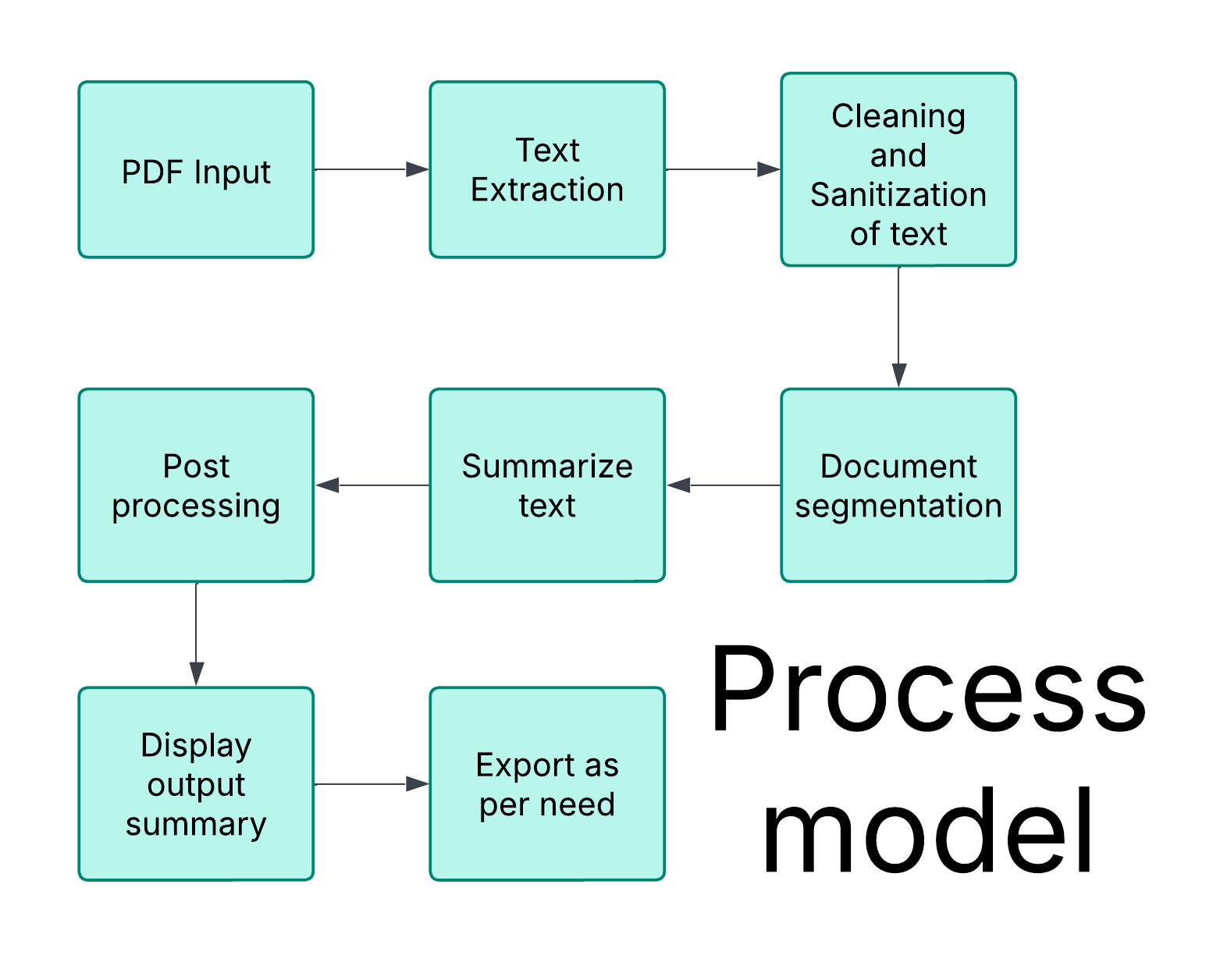
**Minimum Software Requirements**

|  |  |
| --- | --- |
| **Operating System** | Windows/Linux/MacOS (with Python support) |
| **Programming language** | Python 3.10+ |
| **Other tools & tech** | Internet browser |

*Table 3.2 Minimum Software Requirements*

**CHAPTER: 4 PROCESS MODEL**

### CHAPTER 4 PROCESS MODEL

****

*Figure 4.1 Process Model of Project*

## CHAPTER: 5 PROJECT PLAN

### CHAPTER 5 PROJECT PLAN

### Timeline

Week 1: PDF Processing and Extraction

Week 2: Implementing AI Summarization

**Week 3: Testing Robustness**

**Week 4: Optimizing**

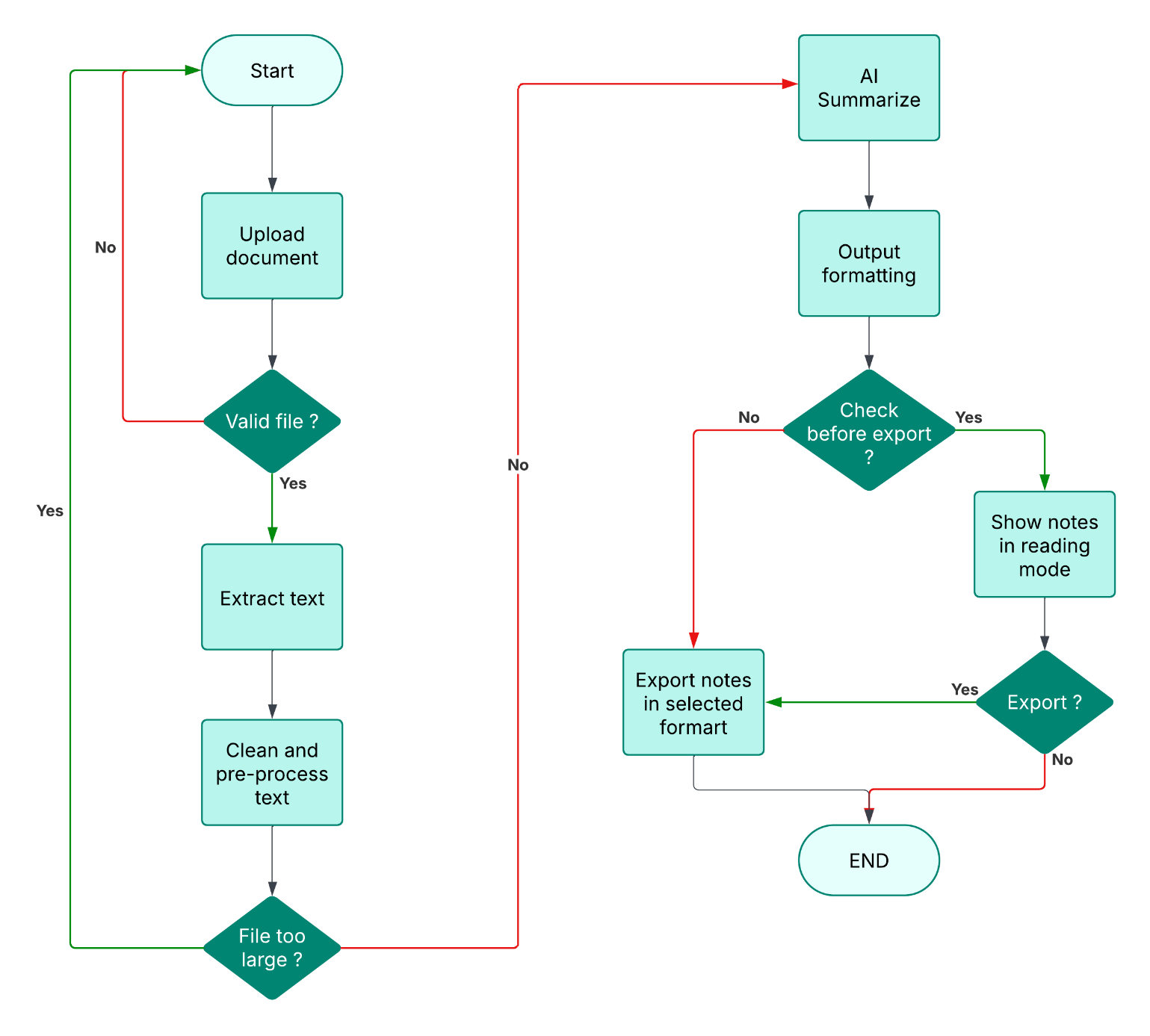
Week 5: Deployment

Web – App : Progressive

# CHAPTER: 6 IMPLEMENTATION DETAILS

### CHAPTER 6 IMPLEMENTATION DETAIL

### Flowchart of Implementation

****

*Figure 6.1 Project Implementation Flowchart*

**CHAPTER: 7 CONCLUSION AND FUTURE WORK**

### CHAPTER 7 CONCLUSION AND FUTURE WORK

**Conclusion**

The **Swift Notes** project aims to implement **AI-power** that automates the extraction, summarization, and presentation of academic content from PDF documents. The project addresses a crucial problem faced by students—**manually summarizing large volumes of study materials**—by providing a fast, accurate, and user-friendly solution. Through the use of advanced Large Language Models (LLMs) like **Google Gemini 1.5 Pro**, the system delivers well-structured, accurate, and topic-diverse study notes that enhance learning efficiency and significantly reduce the time required for manual summarization.

The project is executed through a well-defined process that involves **PDF extraction**, **AI summarization**, **output formatting**, and **exporting** the summarized content. Users interact with the system through a **Django-based web interface**, where they can upload PDF documents, review the generated summaries, and export them to **PDF** or **plain text** formats. This simple yet robust process ensures that students can quickly and effectively generate study materials without spending hours manually compiling notes.

The **performance** of the system is enhanced by leveraging **GPU acceleration** where available. This optimization reduces the time required for processing, making the system responsive and capable of handling multiple documents. Furthermore, by utilizing **Google’s pre-hosted Gemini 1.5 Pro API**, the system provides state-of-the-art summarization capabilities without the need for local model training or heavy computational resources, ensuring both scalability and ease of integration.

The **user experience** is also a focal point of Swift Notes. The web interface is designed to be **intuitive and accessible**, enabling users to navigate seamlessly between uploading documents, viewing summaries, and downloading notes. Additional features such as **text cleaning**, **formatting**, and **export options** contribute to a comprehensive solution that meets the diverse needs of students and academic professionals.

### Future work

While **Swift Notes** aims at achieving its primary goal of providing automated, high-quality study notes; there are several areas for future enhancement and development:

1. **Multilingual Support**: Extend the system’s capabilities to summarize documents in multiple languages beyond English.
2. **OCR Integration**: Implement **Optical Character Recognition (OCR)** to process scanned PDFs and image-based documents (hand-written notes).
3. **Interactive Summarization**: Allow users to interact with and refine summaries by adjusting tone and length.
4. **Batch PDF Processing**: Enable users to upload and summarize **multiple PDFs** simultaneously for greater convenience.
5. **Advanced Customization**: Provide more user control over the summarization process, including options for section-specific summaries and keyword-based extraction.
6. **Improved User Interface**: Enhance the user experience with features like **dark mode**, **search functionality**, and **summary previews**.
7. **Mobile Compatibility**: Optimize the interface for better accessibility and usability on mobile devices.

# CHAPTER: 8 REFERENCES

### CHAPTER 8 REFERENCES

 **Google Gemini API Documentation** – For implementing the Gemini 1.5 Pro model to perform high-quality AI-powered summarization :- <https://ai.google.dev>

 **PyMuPDF (fitz) Documentation** – For extracting and processing text content from PDF documents :- <https://pymupdf.readthedocs.io/>

 **Django Documentation** – For creating the web framework, managing user authentication, routing, and UI rendering :- <https://docs.djangoproject.com/>

 **ReportLab Documentation** – For generating downloadable PDF files from the summarized content :- <https://www.reportlab.com/docs/>

 **OpenAI (ChatGPT)** – Support in conceptualizing project features, refining documentation, and enhancing summarization workflows :- <https://openai.com>

 **Python Official Documentation** – Core reference for implementation using Python libraries and third-party modules :- <https://docs.python.org/>

 **Render / Heroku Docs** – For optional cloud deployment and web hosting services :- <https://render.com/docs>, <https://devcenter.heroku.com/>