

Jaypee Institute of Information Technology, Noida

Department of Computer Science & Engineering and IT



Major Project Title: Kruxx

(A web application for summarizing educational content and generating personalized quizzes)

Enrollment No:

Name of Students:

21103105

Mukund Sarda

21103107

Vansh Gupta

21103110

Aryan Jolly

Name of Supervisor: Dr. Aastha Maheshwari

Course Name: Major Project 2 (15B19CI891)

Program: B. Tech. CS&E

8th Semester

2024-2025

(I)

TABLE OF CONTENTS

Part No.	Description	Page No.
II	Declaration	4
III	Certificate	5
IV	Acknowledgement	6
V	Summary	7
VI	List of Figures	8
VII	List of Symbols and Acronyms	9
VIII	List of Tables	10
Chapter No.	Topics	Page No.
Chapter-1	Introduction	11-19
	1.1 General Introduction	11-13
	1.2 Problem Statement	13-14
	1.3 Significance of the Problem	14-16
	1.4 Empirical Study	16-17
	1.5 Brief Description of Solution Approach	17
	1.6 Comparison of Existing Approaches of the Problem Framed	18-19
Chapter-2	Literature Summary	20-24
	2.1 Summary of Papers Studied	20-23
	2.2 Integrated Summary of the Literature Studied	23-24
Chapter-3	Requirement Analysis and Solution Approach	25-36
	3.1 Overall Description of the Project	25-26

	3.2 Requirement Analysis	26-27
	3.3 Solution Approach	28-33
	3.4 Comparison with Existing Systems	34-36
Chapter-4	Modelling and Implementation Details	37-44
	4.1 Design Diagrams	37-39
	4.2 Implementation Details and Issues	40-41
	4.3 Risk Analysis and Mitigation	42-44
Chapter-5	Testing	45-50
	5.1 Test Plan	45
	5.2 Test Team Details	46
	5.3 Test Schedule	46
	5.4 Test Environment	46
	5.5 Component Decomposition and Type of Testing	47
	5.6 Test Cases	47-48
	5.7 Error and Exception Handling	48
	5.8 Limitations of the Solution	48-50
Chapter-6	Findings, Conclusion and Future Work	51-53
	6.1 Findings	51
	6.2 Conclusion	52
	6.3 Future work	52-53
	REFERENCES	54-55

(II)

DECLARATION

We, **Mukund Sarda, Vansh Gupta, and Aryan Jolly**, hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Mukund Sarda

(21103105)

Vansh Gupta

(21103107)

Aryan Jolly

(21103110)

Place: IIIT, Noida, India

Date: 02 May, 2025

(III)

CERTIFICATE

This is to certify that the work titled “**Kruxx- A web application for summarizing educational content and generating personalized quizzes**” submitted by **Mukund Sarda, Vansh Gupta,** and **Aryan Jolly**, in partial fulfillment for the award of degree of B. Tech in Computer Science & Engineering of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to other University or Institute for the award of this or other degree or diploma.

Dr. Aastha Maheshwari

(Assistant Professor- Senior Grade)

Date: 02 May, 2025

(IV)

ACKNOWLEDGEMENT

We would like to express our sincere gratitude and heartfelt thanks to everyone who has contributed to the successful completion of this major project.

First and foremost, we extend our deepest appreciation to, Dr. Aastha Maheshwari, our project supervisor, for her invaluable guidance, unwavering support, and insightful feedback throughout the duration of this project. Her expertise and encouragement have been instrumental in shaping the project and enhancing its quality.

We would also like to thank Jaypee Institute of Information Technology for providing the necessary resources and conducive environment for conducting this project. The library and laboratory facilities have played a crucial role in the research and development process.

We would like to thank our families and friends for their continuous encouragement and understanding during the challenging phases of this project. Their belief in our abilities has been a driving force behind our perseverance.

This project has been a significant learning experience, and we are truly grateful to have had the opportunity to work on it. Thank you all for being an integral part of this journey.

Mukund Sarda

(21103105)

Vansh Gupta

(21103107)

Aryan Jolly

(21103110)

Date: 02 May, 2025

(V)

SUMMARY

Our project titled “Kruxx” has been developed as a comprehensive solution to tackle the many challenges faced by students and educators in today’s fast-paced academic world. It brings together a wide range of technologies, including Natural Language Processing (NLP) and Optical Character Recognition (OCR), and it’s primarily built using Flask. It integrates powerful tools and libraries like textract, YouTube Transcript API, Google Translate and Google Generative AI. These technologies come together to create a seamless platform aimed at simplifying the processes of learning, content summarization, and quiz generation.

Kruxx enables users to upload educational material across multiple formats; text documents like PDFs and DOCs, presentation files like PPTs, scanned images of handwritten notes or textbooks or video links from sources like YouTube. Using models like BART and BERT, it condenses large volumes of content into crisp, meaningful summaries while carefully retaining the essence of the material. It also dynamically generates quizzes from the given material, offering users multiple-choice questions tailored to the content they uploaded. It also can detect if a YouTube video is clickbait and even gives you recommendations based on the video.

The primary goal of Kruxx is simple yet ambitious: to revolutionize the way people engage with educational content. By reducing the time needed to go through vast amounts of study material, promoting active learning through quizzes, and offering a tool that is flexible enough for students and teachers, Kruxx positions itself as a game-changer in the realm of educational technology.

Mukund Sarda
(21103105)

Vansh Gupta
(21103107)

Aryan Jolly
(21103110)

Dr. Aastha Maheshwari

Date: 02 May, 2025

(VI)

LIST OF FIGURES

Figure	Page No.
Figure 3.1 Homepage	31
Figure 3.2 OCR and Summarisation Menu for Image	31
Figure 3.3 Summarisation Menu for PDF (PPT/DOC)	32
Figure 3.4 Recommendation of YouTube Link	32
Figure 3.5 Clickbait Detector for YouTube Videos	33
Figure 3.6 Quiz Section of Kruxx	33
Figure 4.1 Use Case Diagram	37
Figure 4.2 Class Diagram	37
Figure 4.3 Control Flow Diagram	38
Figure 4.4 Sequence Diagram	39

(VII)

LIST OF SYMBOLS AND ACRONYMS

Symbols	Meaning
BART	Bidirectional Auto-Regressive Transformers
BERT	Bidirectional Encoder Representations from Transformers
API	Application Programming Interface
OCR	Optical Character Recognition
NLP	Natural Language Processing
PDF	Portable Document Format
AI	Artificial Intelligence
HTML	Hypertext Markup Language
USP	Unique Selling Proposition
PPT	PowerPoint Presentation
DOC	Document

(VIII)

LIST OF TABLES

S. No.	Description	Page No.
1	Summarization	18
2	Quiz Generation	18
3	Unified Workflow	18
4	Summary of Papers Studied	20-23
5	Feature Comparison Table	35
6	Risk Identification	42
7	Risk Area Analysis	43
8	Mitigation Approaches	43-44
9	Test Plan	45
10	Test Team Details	46
11	Test Schedule	46
12	Test Environment	46
13	Component Decomposition and Type of Testing	47
14	Test Cases	47-48
15	Error and Exception Handling	48

CHAPTER- 1

INTRODUCTION

1.1) GENERAL INTRODUCTION

In today's fast-moving world where information is endless and attention is limited, students and educators find themselves caught in the challenge of managing massive volumes of educational content. Whether it is preparing for competitive exams, revising crucial course material, or trying to teach students in a meaningful way, the task of processing, summarizing, and truly understanding information has become overwhelming and time-consuming. The need for a smarter, quicker, and more effective approach to learning has never been greater.

A modern, web-based platform, Kruxx is designed to bridge the gap between traditional learning methods and new-age technological advancements. By combining Natural Language Processing (NLP), Optical Character Recognition (OCR), and the best practices of web development, Kruxx changes the way users interact with educational material. From summarizing endless study notes to generating personalized quizzes to getting YouTube videos recommendations to checking whether YouTube videos are clickbait or not, Kruxx not only boosts productivity but also encourages active engagement and ensures better retention of knowledge. In a world where learning often feels like a race against time, Kruxx stands out as a much-needed companion on the journey towards smarter education.

Why Kruxx?

Education today is not just about textbooks anymore. It includes research papers, elaborate presentations, video lectures, podcasts, and a dozen other forms of content. Yet, despite this abundance, students often find themselves struggling, and the reasons are more familiar than we realise:

1. **Information Overload:** With so much material floating around, students end up drowning in information. The real challenge is not accessing knowledge, but extracting the key concepts and understanding what truly matters. Important ideas often get lost in the chaos, leading to frustration and inefficiency.

2. **Limited Time:** Exams do not wait. Assignments do not wait. And students do not have the luxury of endless time to sift through mountains of content. They are caught in a constant race against deadlines, leaving them little room to process and retain what they are learning.
3. **Lack of Personalized Assessment:** Not every student thinks the same way. Yet, most quizzes and evaluations today assume they do. Generic assessments fail to identify individual learning gaps, leaving students unaware of where they need to focus and improve.
4. **Misleading content:** Nowadays, a lot of misinformation is present on the internet. This is prevalent even in the case of YouTube where people may post ‘Clickbait’ videos which have a misleading title and thumbnail to lure the viewers into clicking on the video but the actual content in them is different.

Educators face their own battles. They are expected to create personalised quizzes, analyse each student's performance, and tweak their teaching strategies all while handling a classroom full of unique individuals, each with different needs and learning styles. It is a lot to ask for, and often, traditional methods just do not cut it.

That’s exactly where Kruxx comes into play. By combining summarisation and quiz generation into one single platform, Kruxx doesn’t just solve these problems, it redefines the learning experience. It helps students focus better, learn smarter, and gives educators the insights they need to teach more effectively.

Core Components

1. **Content Summarization:** Kruxx uses advanced NLP models like BART and BERT to generate summaries that keep the essential information intact while removing any redundant details. Whether it’s abstractive summarization (rephrasing) or extractive summarization (key sentence extraction), the platform caters to both needs with precision.
2. **Quiz Generation:** Based on the summarized content, Kruxx dynamically creates quizzes that maintain contextual relevance. The quizzes are designed with varying levels of difficulty to suit users with different levels of understanding, making the learning experience more targeted and effective.
3. **OCR Integration:** For users who depend on physical study materials like handwritten notes or scanned documents, Kruxx integrates advanced OCR tools like EasyOCR and

DocTR. This ensures that non-text inputs are converted efficiently into editable text, making the platform versatile across different types of study resources.

4. **Clickbait Checker Integration:** Kruxx comes equipped with an intelligent Clickbait Detection system to help users quickly judge the authenticity of YouTube videos. By combining the YouTube Data API, transcript analysis, comment sentiment evaluation, and advanced NLP techniques like cosine similarity and polarity scoring, Kruxx makes it easier to separate genuine educational content from clickbait. This way, users can invest their time wisely and focus on trustworthy resources.
5. **Video Recommendation System:** Kruxx elevates the learning experience by offering personalized video recommendations, seamlessly integrating with tools like yt-dlp and YouTube APIs. By analysing video titles, views, likes, and filtering out repetitive content, Kruxx ensures that users are directed towards the most relevant, high-quality study videos. This makes discovering content faster, smarter, and perfectly aligned with each learner's unique needs.

Kruxx is not just another tool — it is a complete learning assistant built to simplify and elevate the educational journey. By bringing together automation and cutting-edge AI capabilities, Kruxx changes the way users engage with educational content, helping them build smarter study habits and boost their academic performance. With its versatile features and focus on user needs, Kruxx proves to be a valuable resource not just for students, but also for educators and institutions alike.

1.2) PROBLEM STATEMENT

In today's educational world, both students and professionals face a major challenge: efficiently preparing for exams or skill assessments amidst a sea of study material. This content is spread across different formats—text, PDFs, PPTs and video lectures — making it tough to focus on what truly matters. The issue is made worse by several key factors:

1. **Information Overload:** The massive amount of data in textbooks, online resources, and lecture notes often leads to confusion, procrastination, and inefficient study habits.
2. **Lack of Personalization:** Most study tools don't address individual learning needs. While students vary in their understanding, most resources offer generic content that doesn't help them improve where they need it most.

3. **Limited Time:** With exam preparations often coming down to the wire, students need a concise, yet thorough, way to review materials and make the most of the time they have left.
4. **Inadequate Self-Assessment Tools:** Knowing what you've mastered is crucial, but many platforms fail to offer personalized, adaptive quizzes that match the study material, leaving students uncertain about their progress.
5. **Impact of Solving the Problem:** The successful rollout of Kruxx will completely transform how students approach exam preparation by tackling the core challenges of summarization and assessment. It will drastically cut down study time, boost comprehension, and lead to better learning results, ultimately making education more efficient and accessible. This initiative ties into the larger mission of democratizing education, offering smart tools that adapt to each learner's unique needs.

1.3) SIGNIFICANCE OF THE PROBLEM

The education landscape has seen a dramatic shift in recent years, driven by technological advancements, the growth of online learning platforms, and an explosion of available resources. While these changes have made knowledge more accessible, they've also brought about a major challenge: information overload. Both students and educators are now faced with the overwhelming task of processing, understanding, and retaining vast amounts of content in a limited time.

The significance of the problem lies in addressing the following critical challenges:

1. **Managing Information Overload:** With an abundance of resources like textbooks, research papers, videos, and podcasts, students face the daunting task of:
 - Filtering relevant information from an overwhelming sea of content.
 - Identifying key concepts and areas that need focus.
 - Managing time efficiently for studying, revising, and preparing for assessments.
2. **Lack of Summarization Tools for Diverse Formats:** Though text summarization tools exist, they typically focus on formats like plain text or PDFs, leaving out other popular formats such as:
 - Scanned notes or handwritten documents.

- Video lectures, especially from platforms like YouTube.
3. **Absence of Personalized Assessment:** Current assessments fail to adapt to individual learning needs, with issues like:
- Generic assessments: Quizzes that don't address the unique strengths or weaknesses of students.
 - One-size-fits-all approach: Students with different levels of understanding are evaluated on the same terms, leading to inaccurate assessments.
4. **Time Constraints in Exam Preparation:** The growing complexity of curricula, combined with limited prep time, creates immense pressure. Key challenges include:
- Difficulty in organizing and prioritizing materials.
 - Lack of time for revision due to long, unstructured content.
 - The mental stress of managing vast resources, affecting productivity and focus. By automating summarization and quiz generation, students save time, allowing them to concentrate on mastering the material.
5. **Challenges for Educators:** Teachers face challenges such as:
- Time-consuming quiz creation: Developing quizzes that align with course material is repetitive and tedious.
 - Difficulty in evaluating performance: Manual analysis of results is inefficient, especially for large classes.
 - Lack of summarization tools: Condensing textbooks, research papers, and lecture notes into digestible formats for students is often a struggle. These issues prevent educators from focusing on what matters most: effective teaching and mentoring.
6. **Inefficiencies in Traditional Learning Methods:** Traditional methods often rely on:
- Reading and highlighting from textbooks or notes.
 - Writing summaries or creating flashcards for revision.
 - Designing practice questions for self-assessment.

These approaches are time-consuming, error-prone, and fail to utilize the power of technologies like Natural Language Processing (NLP) and automation to improve learning efficiency.

1.4) EMPIRICAL STUDY

The empirical study conducted as part of the Kruxx project sought to examine the real-world challenges that students and educators face when preparing for exams and handling large volumes of educational content. The study provided valuable insights into the shortcomings of current tools, user expectations, and the potential of integrating advanced technologies like NLP and OCR into a cohesive platform.

The study highlighted key pain points and areas where improvements can be made in existing systems:

Challenges Faced by Students

1. **Information Overload:** 57% of students struggle with managing the overwhelming study material, especially during exam time, as they find it hard to prioritize key concepts without proper summarization tools.[1]
2. **Time Constraints:** 66% of students feel pressed for time when summarizing notes, textbooks, or lectures and often skip material because they cannot afford to spend the time on it.[2]
3. **Limited Assessment Tools:** While 78% of students use online quizzes for self-assessment, they often find them too generic and not tailored to their specific needs.[3]
4. **Diverse Input Formats:** Students rely on handwritten notes, scanned documents, and video lectures, but current tools do not efficiently process these different formats.

Challenges Faced by Educators

1. **Time-Consuming Quiz Creation:** 62% of educators find grading to be the worst part of their job.[4]
2. **Performance Tracking:** Educators lack effective tools to track student performance and offer targeted feedback.

Existing Tool Limitations

1. **Narrow Focus:** Tools like SMMRY or Resoomer only handle plain text, neglecting formats like PDFs or video transcripts.

2. **Fragmented Solutions:** No single platform combines both summarization and quiz generation, forcing users to rely on multiple tools.

1.5) BRIEF DESCRIPTION OF SOLUTION APPROACH

The **Kruux** is built as a modular web application, where each part handles a specific task, ensuring it's scalable, easy to maintain, and user-friendly. The system is made up of several key components:

1. **Frontend:** Designed with HTML, Tailwind CSS, React, and JavaScript, it offers a responsive interface where users can upload files, choose their summarization, or quiz preferences, and view the results with ease.
2. **Backend:** Powered by Flask, a lightweight Python framework, the backend manages user requests, processes data, and generates the necessary outputs. It acts as the bridge between the frontend and various processing modules, such as summarization, OCR, and quiz creation.
3. **Processing Modules:** These modules leverage advanced NLP models to summarize content and generate quizzes based on those summaries. They also extract text from images or scanned documents for further analysis.
4. **APIs and External Tools:** The system integrates with tools like the YouTube Transcript API for processing video content and uses libraries like EasyOCR for extracting text from images and non-text formats.

Advantages of the Solution Approach

1. **Automation:** By automating the summarization and quiz creation process, the solution saves considerable time and effort for both students and educators.
2. **Personalization:** The system offers customized summaries and quizzes that cater to the unique needs of each user.
3. **Accessibility:** It accommodates a wide variety of input formats, ensuring that it's accessible to users with different resources and preferences.
4. **High Accuracy:** The use of cutting-edge NLP models and OCR technology guarantees accurate summarization and content extraction, delivering reliable results.

1.6) COMPARISON OF EXISTING APPROACHES TO THE PROBLEM FRAMED

Kruxx overcomes the limitations of existing tools and manual methods by integrating advanced technologies into a single, comprehensive platform.

Summarization

Feature	Existing Tools	Kruxx
Input Format Support	Plain text	PDF, DOC, PPT, images, YouTube video, website links
Summarization Type	Extractive only	Extractive and Abstractive
Customization	Limited	Customizable length and style
NLP Model Integration	Basic	Advanced models (BART, BERT)
Multimedia Support	Not supported	Fully supported

Quiz Generation

Feature	Existing Tools	Kruxx
Automation	Manual input required	Fully automated from input content
Contextual Relevance	User-dependent	Ensures alignment with uploaded material
Time Efficiency	Time-consuming	Faster

Unified Workflow

Feature	Existing Approaches	Kruxx
Workflow Integration	Fragmented	Unified platform for all features
User Interaction	Multiple tools needed	Single interface for all processes
Accessibility	Varies	Cross-platform and user-friendly

Why Kruxx is Better?

1. **Time Efficiency:** Automates summarization and quiz creation, cutting down the time needed for these tasks by a significant margin.
2. **Comprehensive Features:** Handles a variety of input formats, from text to images to websites/YouTube links to different formats like PPTs, DOCs and PDFs, offering flexibility and inclusivity.
3. **Personalized Learning:** Creates quizzes tailored to each user's needs, providing performance insights and suggestions for improvement.
4. **Advanced Technology Integration:** Uses cutting-edge NLP models like BART and BERT, along with OCR tools, to ensure accurate and contextually relevant results.
5. **User-Centric Design:** Combines an intuitive, responsive interface with robust backend processing, ensuring a smooth and seamless experience for users.

CHAPTER- 2

LITERATURE SURVEY

2.1) SUMMARY OF PAPERS STUDIED

The following table summarizes the research papers reviewed during the development of **Kruxx**. Each paper contributed to understanding specific aspects of summarization, quiz generation, and the integration of NLP technologies into educational systems.

Author(s)	Title	Publish Date	Summary	Reference
Deepika Varshney, Dinesh Kumar Vishwakarma	A unified approach for detection of Clickbait videos on YouTube using cognitive evidences	2021	This paper introduces a Clickbait Video Detector (CVD) that aims to spot clickbait videos on YouTube by pulling in three layers of cognitive clues—what the video says (like how much the speech matches the title and the number of dislikes), what the viewers are saying (through comment analysis), and who’s posting it (by checking the channel’s reputation). It picks up on these features, blends them together, and then uses machine learning models to figure out if a video is clickbait or not. And the best part? It beats the accuracy of the methods already out there.	[5]
Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed,	BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation,	2019	This paper rolls out BART, a transformer-based model built specifically for abstractive summarization. It breaks down how BART cleverly mixes bidirectional and autoregressive tricks to tackle long chunks of text and churn out crisp,	[6]

Omer Levy, Ves Stoyanov, Luke Zettlemoyer	Translation, and Comprehension		concise summaries. The paper also walks through the pre-training tasks that helped fine-tune BART, making it a solid choice for the summarization tasks tackled in this project.	
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	2019	The authors dive into BERT, a bidirectional transformer model that's been a game-changer for all sorts of NLP tasks. They lay out how BERT is great at pulling meaningful features out of text, which makes it a natural fit for extractive summarization. Its knack for holding onto contextual information turned out to be crucial when it came to building the extractive summarization part of Kruxx.	[7]
Jingqing Zhang, Yao Zhao, Mohammad Saleh, Peter J. Liu	Pegasus: Pre- training with Extracted Gap- sentences for Abstractive Summarization	2020	This paper takes a look at Pegasus, a pre-trained NLP model that's geared toward abstractive summarization. It brings something new to the table with a unique pre-training objective that makes summarizing large documents a lot more effective. The ideas behind Pegasus were also kept in mind when sizing up models like BART to figure out what would work best for Kruxx.	[8]
Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon	What is Wrong with Scene Text Recognition Models and Data?	2019	The authors dig into the hurdles faced by OCR technology and suggest ways to make text recognition better, especially in tricky environments. Their findings played a key role in picking the right OCR tools—like EasyOCR and DocTR—for pulling	[9]

Han, Sangdoo Yun, Seong Joon Oh, Hwalsuk Lee			text out of handwritten notes and scanned documents while building Kruxx.	
Zachary C. Lipton, John Berkowitz, Charles Elkan	A Critical Review of Recurrent Neural Networks for Sequence Learning	2015	This paper looks back at how Recurrent Neural Networks (RNNs) were used for sequence-based tasks like summarization. Even though models like BERT and BART have since left RNNs in the dust, the paper gives some valuable context on how sequence modeling has evolved—and how today's methods have stepped up their game compared to the old-school approaches.	[10]
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin	Attention Is All You Need	2017	The authors dive into how attention mechanisms are used in deep learning models and how they've been a game-changer for summarization tasks. These mechanisms had a direct hand in making models like BART and BERT way more effective at handling long text inputs, which turned out to be a big win for the summarization features in Kruxx.	[11]
Vatsal Raina, Mark Gales	Multiple-Choice Question Generation: Towards an Automated	2022	This study takes a deep dive into a fully automated system that generates multiple-choice questions (MCQs) straight from context paragraphs. It brings in new performance criteria—like grammatical accuracy,	[12]

	Assessment Framework		answerability, diversity, and complexity—to assess the quality of the MCQs, tackling some of the issues with traditional evaluation methods.	
--	----------------------	--	--	--

2.2) INTEGRATED SUMMARY OF LITERATURE STUDIED

The research papers reviewed provide a strong foundation for developing the **Kruxx** platform. The integrated insights from these papers are summarized below:

1. Summarization Models

The papers on BART, BERT, and Pegasus reflect a clear picture of how summarization tech has evolved over time. BERT shines when it comes to extractive summarization, picking out and holding onto the key sentences. On the other hand, BART and Pegasus are the go-tos for abstractive summarization, where the focus is on rephrasing content into something more concise. These insights helped shape how both summarization techniques were brought into Kruxx, giving users the freedom to choose the one that best fits their needs.

2. Optical Character Recognition (OCR)

The research on OCR, like Baek et al.'s work, really underscored the need for robustness in text recognition. By tapping into the power of EasyOCR and DocTR, Kruxx brings in cutting-edge OCR technology, making sure that text extraction from scanned documents and handwritten notes is spot-on and reliable.

3. Quiz Generation

The work by Raina et al. really highlights the expanding role of AI in education. By using pre-trained models for question generation, Kruxx ensures that quizzes are always on point with the content. These ideas were directly woven into Kruxx to make quizzes more relevant

4. General NLP Advancements

The work by Vaswani et al. on attention mechanisms shed light on how deep learning techniques, particularly transformers, have completely changed the game for text processing. This opened the door for using advanced models like BART and BERT in Kruxx, making summarization and question generation more efficient and effective.

5. Clickbait Video Detection

The work by Varshney et al. analyse YouTube video titles, transcripts, and comments to detect potential clickbait using the YouTube Data API, speech-to-text, comment sentiment evaluation, and advanced NLP techniques like cosine similarity and polarity scoring. This helped Kruxx integrate this functionality for the YouTube video link section so that users can rest assured that they are spending their time on useful and informative videos and not clickbait videos.

CHAPTER- 3

REQUIREMENT ANALYSIS AND SOLUTION APPROACH

3.1) OVERALL DESCRIPTION OF THE PROJECT

Key Features

1. **Multi-Format Input Support:** Accepts various formats like PDFs, DOCs, texts, scanned notes (images), video links (like YouTube), and website links. Converts multimedia inputs into readable text using OCR.
2. **Advanced Summarization:** Creates concise summaries by rephrasing the input while keeping the original meaning intact (using the BART model). It can also extract the most relevant sentences directly (using the BERT model). Offers options to customize the summary length—short, medium, or long.
3. **Dynamic Quiz Generation:** Automatically generates MCQs based on the given material. Uses generative AI to keep the questions contextually relevant.
4. **Seamless Text-to-Speech Conversion:** Converts the generated summaries into natural-sounding speech using the speech Synthesis property of JavaScript. Offers adjustable speaking speeds and different voices for different languages.
5. **Multi-Language Translation:** Instantly translates the summarized content into multiple languages using Google Translate APIs. Users can select from a wide range of supported languages based on their preferences.
6. **Clickbait Detection for YouTube Links:** Analyzes YouTube video titles, transcripts, and comments to detect potential clickbait using the YouTube Data API, transcript analysis, comment sentiment evaluation, and advanced NLP techniques like cosine similarity and polarity scoring.
7. **Smart Video Recommendation System:** Generates intelligent video recommendations based on the input YouTube link using yt-dlp for metadata extraction and YouTube APIs for content analysis. Ensures that recommended videos are high-quality, contextually relevant, and aligned with user interests.
8. **User-Friendly Interface:** A clean, intuitive web platform that's easy to navigate.

Unique Selling Points (USPs)

- **All-in-One Platform:** Combines summarization, quiz generation, clickbait checker and YouTube recommendations in a single application.
- **Advanced AI Models:** Leverages state-of-the-art transformer models and generative AI API to deliver superior performance.
- **Versatile Input Handling:** Processes text, images, videos, and other document formats seamlessly.
- **Customization:** Offers personalized settings for summaries and quizzes, ensuring user-specific outcomes.

3.2) REQUIREMENT ANALYSIS

Functional Requirements

1. **Content Upload:** Users should be able to upload educational material in various formats such as PDFs, DOCs, PPTs, scanned images, video URLs (like YouTube lectures), and website links. The system must validate the file types and display clear error messages for unsupported formats.
2. **Content Summarization:** The platform should offer both abstractive summarization (for natural, rephrased summaries) and extractive summarization (for picking key sentences). Users should also have the option to choose the summary length—short, medium, or long.
3. **Quiz Generation:** Based on the given content, the system should automatically create multiple-choice questions (MCQs) with well-designed distractors to add complexity and depth to the quizzes.
4. **User Interface:** A clean and intuitive interface must be provided for uploading content, viewing summaries, and taking quizzes, with smooth functionalities.
5. **API Integrations:** The system should use external tools like the YouTube Transcript API for extracting text from video lectures and other APIs like gemini-api and YouTube Data API for required purposes.

Non-Functional Requirements

1. **Performance:** The system should be able to process uploaded content and generate summaries within a few seconds for small to medium-sized inputs (for example, PDFs under 10 pages). Quiz generation should be completed within a few seconds for standard content to ensure a smooth user experience.
2. **Usability:** The platform should be easy to navigate for users of all technical backgrounds and clear error messages should be provided to assist users throughout their interaction with the system.
3. **Reliability:** The system should ensure high reliability with minimal downtime. A robust error-handling mechanism should be in place to allow recovery from unexpected failures without affecting user experience.
4. **Maintainability:** A modular architecture must be used to make updates and maintenance straightforward. Detailed documentation should be provided for both developers and administrators to ensure smooth future enhancements.

Technical Considerations

1. NLP Models:

- Both BART and BERT models demand a lot of computational power for summarization tasks, so the system must be optimized to prevent delays.
- Pre-trained models should be used to cut down on both development time and computational expenses.

2. OCR Tools:

- Tools like EasyOCR and DocTR can struggle with recognizing highly distorted or low-resolution text. To address this, the system should preprocess images to enhance OCR accuracy.

3. Infrastructure:

- The platform must be compatible with standard hardware and software setups.

3.3) SOLUTION APPROACH

The Kruxx platform leverages advanced technologies like Natural Language Processing (NLP), Optical Character Recognition (OCR), and smart quiz generation to create a modular, scalable web-based solution. This approach is crafted to:

1. Automate Time-Consuming Tasks:

- Streamline the process of summarizing large amounts of content and generating quizzes, saving time for users.

2. Support Multiple Input Formats:

- Ensure compatibility with various content types, including text, images, video, and website links, making the platform accessible to diverse users.

3. Provide Real-Time Result:

- Offer immediate result to enhance learning and keep users on track.

Architectural Design

The Kruxx platform is built with a modular architecture, ensuring that each component operates independently for better scalability, ease of maintenance, and seamless integration. The architecture is structured into three core layers:

1. Frontend Layer:

- Designed using HTML, Tailwind CSS, React, and JavaScript, the frontend provides a responsive and user-friendly interface.
- Features dynamic forms for uploading files, selecting summarization preferences, generating quizzes, and visualizing results.

2. Backend Layer:

- Built with Flask, a lightweight Python framework, the backend handles user requests and coordinates processing tasks.
- It serves as the bridge between the frontend and the core processing modules, overseeing session management, data storage, and API interactions.

3. Data Processing Layer:

- This layer handles the critical tasks of text summarization, OCR-based text extraction, and quiz generation.
- It integrates advanced NLP models (like BART and BERT) and OCR tools (such as EasyOCR and DocTR) for precise processing.
- It also provides APIs for processing multimedia content like YouTube videos links.

Step-by-Step Workflow

The Kruxx platform's solution approach ensures a seamless flow from user input to output generation through the following steps:

Step 1: User Input

- Users engage with the web app to upload study materials or provide URLs for video files.
- Supported input formats include:
 - **Text files:** PDF, DOC, PPTs and TXT.
 - **Images:** Scanned notes, handwritten content, or textbook pages.
 - **Videos:** YouTube links, which are processed to extract transcripts.
 - **Links:** Website links which are processed to extract text.

Step 2: Preprocessing

Before the core tasks like summarization or quiz generation, the content undergoes preprocessing to enhance compatibility and quality:

1. Text Preprocessing:

- Remove unnecessary characters, special symbols, and formatting inconsistencies.
- Tokenize the text into smaller units for easier processing.

2. Image Processing:

- Use libraries like PIL to improve image quality for better OCR results.
- Extract text using EasyOCR or DocTR.

3. Video Transcription:

- Convert video to text using tools like YouTube Transcript API.

Step 3: Content Summarization

After preprocessing, the system generates concise summaries:

1. Abstractive Summarization:

- Powered by BART, this method rephrases text to create human-like summaries while keeping the core meaning.
- Ideal for creative and context-rich summaries.

2. Extractive Summarization:

- Powered by BERT, it identifies and extracts key sentences directly from the text.
- Perfect for maintaining the original phrasing and structure of the content.

Users can choose between abstractive or extractive summarization and specify the desired length (short, medium, or long).

Step 4: Quiz Generation

The quiz generation module dynamically creates MCQs based on the given content:

1. Text Extraction:

- Text is extracted from the inputted file/link

2. API Calling

- Generative AI API is called to generate and fetch the quiz

Step 5: User Interface and Output Delivery

- Summaries, quizzes, and performance results are displayed on an intuitive, user-friendly interface.
- YouTube videos are recommended so that user can enhance its knowledge about the topic.

Website UI:

Below attached are screenshots of the homepage, website interface and different functionalities of our project.

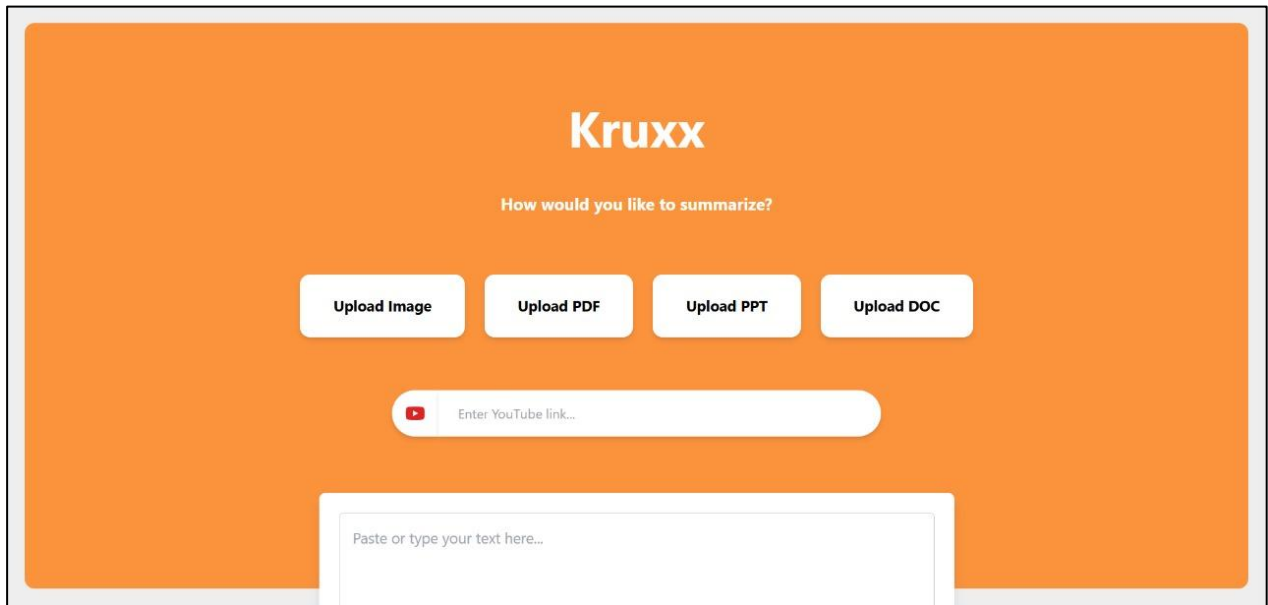


Fig 3.1- Homepage

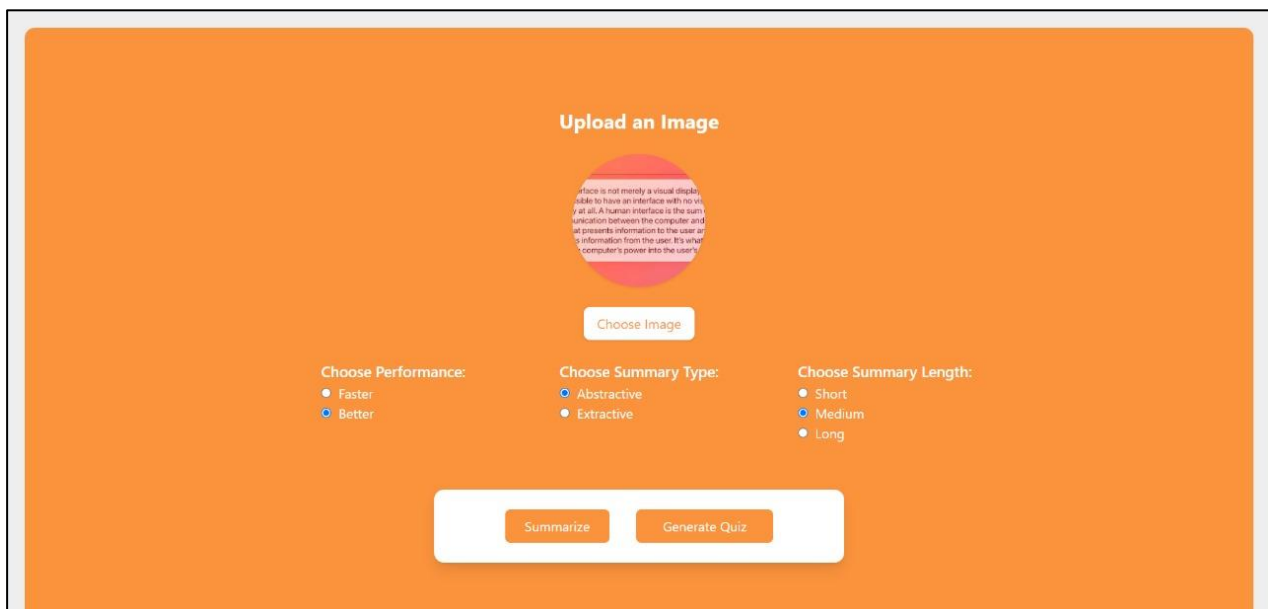


Fig 3.2- OCR and Summarisation Menu for Image

Upload a PDF

major project synopsis.pdf

Choose PDF

Choose Summary Type:

☒ Abstractive

☐ Extractive

Choose Summary Length:

☐ Short

☒ Medium

☐ Long

Summarize Generate Quiz

Fig 3.3- Summarisation Menu for PDF (PPT/DOC)

https://youtube.com/watc Summarize Hide Recommendations Check Clickbait

Recommended Videos

Semester Exam DBMS IN 5 HOURS

Complete DBMS Data Base Management System in one shot | Semester Exam | Hindi

Length: 333 min 30 sec

Channel: KnowledgeGATE by Sanchit Sir

Watch Video

Exam fear Vs Gate Smashers #shorts#ytshorts #trending #viral #gatesmashers#trendingshorts

Length: 0 min 13 sec

Channel: Gate Smashers

Watch Video

Resources to master DBMS | # shorts

Length: 0 min 20 sec

Channel: Yashashvi Bhadauria

Watch Video

DBMS Syllabus Discussion | Important Topics for DBMS | Gate 2022 Exam

Length: 8 min 23 sec

Channel: GATE AT ZEAL

Watch Video

Fig 3.4- Recommendations of YouTube Link

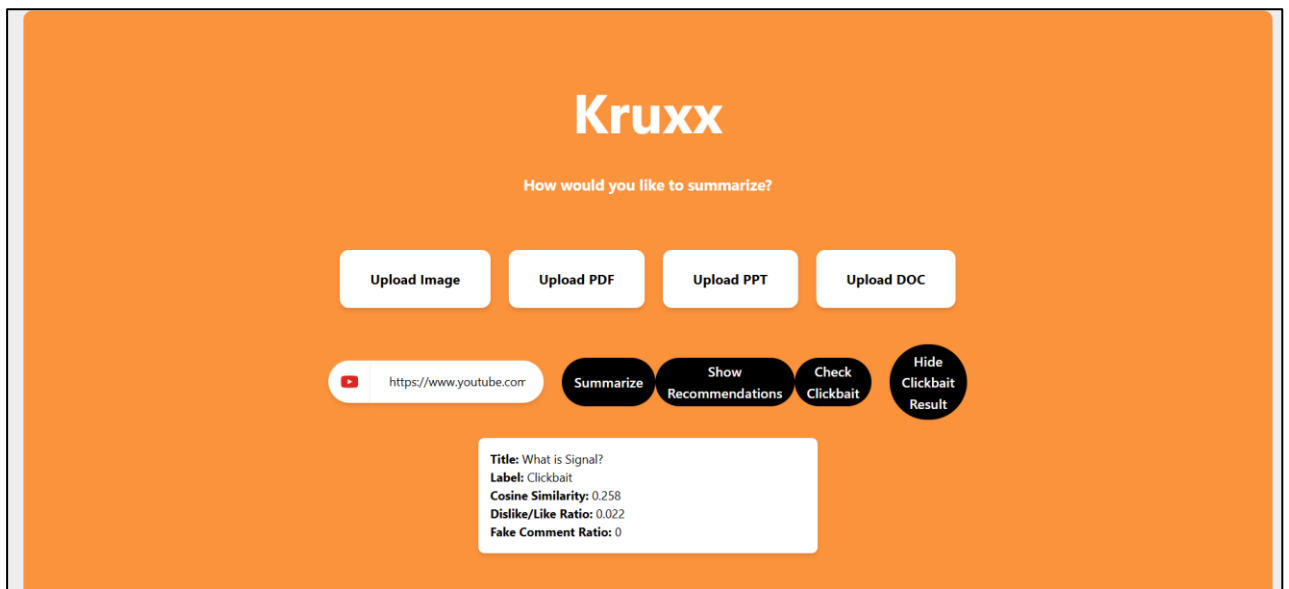


Fig 3.5- Clickbait Detector for YouTube Videos

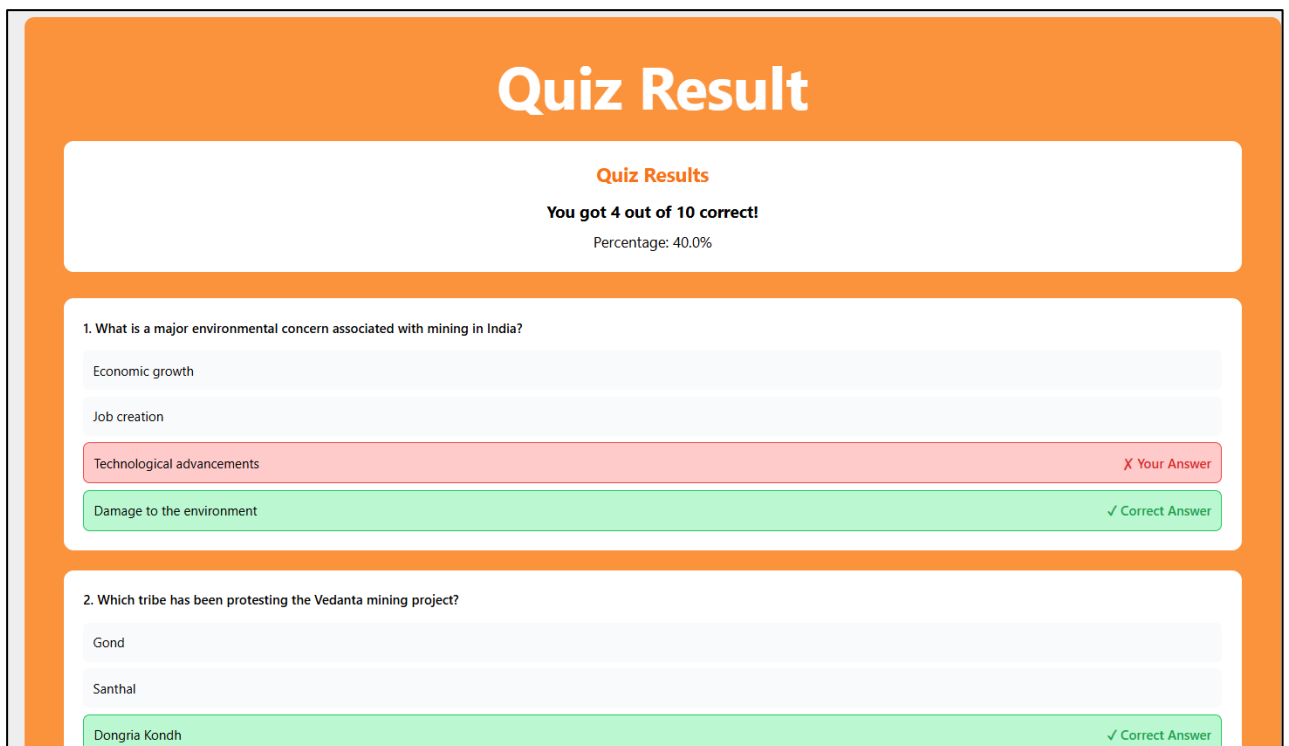


Fig 3.6- Quiz Section of Kruxx

3.4) Comparison with Existing Systems

This section dives deep into how Kruxx stacks up against other similar systems, examining its features, functionalities, and performance. The aim is to spotlight what makes Kruxx stand out and show how it outperforms other tools when it comes to tackling the challenges of personalized exam preparation.

1. Existing Systems

Several platforms offer partial or similar functionalities to **Kruxx**, including content summarization tools, quiz generation systems, and learning management systems (LMS). Below are key competitors:

1. QuillBot:

- **Functionality:** Primarily focuses on paraphrasing and summarization.
- **Strengths:**
 - Effective at rephrasing text.
 - User-friendly and quick processing.
- **Limitations:**
 - Does not support input formats like videos or images,
 - Lacks quiz generation.

2. Kahoot!

- **Functionality:** Primarily a quiz-based platform for educators and students.
- **Strengths:**
 - Interactive and engaging quizzes.
 - Real-time multiplayer features.
- **Limitations:**
 - No summarization capabilities.
 - Relies on manual quiz creation by educators.

3. Edmodo:

- **Functionality:** Learning management system with classroom collaboration tools.
- **Strengths:** Facilitates classroom communication and resource sharing.
- **Limitations:**
 - Lacks content summarization and automated quiz generation.
 - Limited to text-based inputs.

4. SummarizeBot:

- **Functionality:** Provides automated summarization for text, PDFs, and web pages.
- **Strengths:** Supports multiple input formats.
- **Limitations:**
 - No quiz generation.
 - Summaries lack customization for length or type.

Feature Comparison Table

Feature	Kruxx	QuillBot	Kahoot!	Edmodo	SummarizeBot
Content Summarization	Yes (BERT/BART)	Yes	No	No	Yes
Customizable Summaries	Yes (Short/Med/Long)	Limited	No	No	Limited
Quiz Generation	Yes (MCQs)	No	Yes (Manual)	No	No
Input Formats	Text, PDFs, DOCs, PPTs, Images, YouTube Videos/website links	Text only	None (Quiz Only)	Text only	Text, PDFs
Clickbait Detection	Yes	No	No	No	No

2. Key Differentiators of Kruxx

1. All-in-One Functionality:

- It brings together summarization, quiz creation, clickbait detection and YouTube videos recommendation all in one place, saving you the hassle of juggling multiple tools.

2. Versatile Input Support:

- Whether it's text, images, videos, or website links, this platform can handle it all, making it perfect for diverse learning materials.

3. Tailored Summaries:

- You can get both extractive (BERT) and abstractive (BART) summaries, with options for different lengths—short, medium, or long—depending on what you need.

4. Smart Quiz Creation:

- It generates MCQs automatically from the given text, and even includes relevant but tricky distractors to keep things interesting.

CHAPTER- 4

MODELLING AND IMPLEMENTATION DETAILS

4.1) DESIGN DIAGRAMS

The Use Case Diagram helps in visualizing the different functionalities that our project provides. It also represents the interactions between the user, APIs, and the core functionalities of the project.

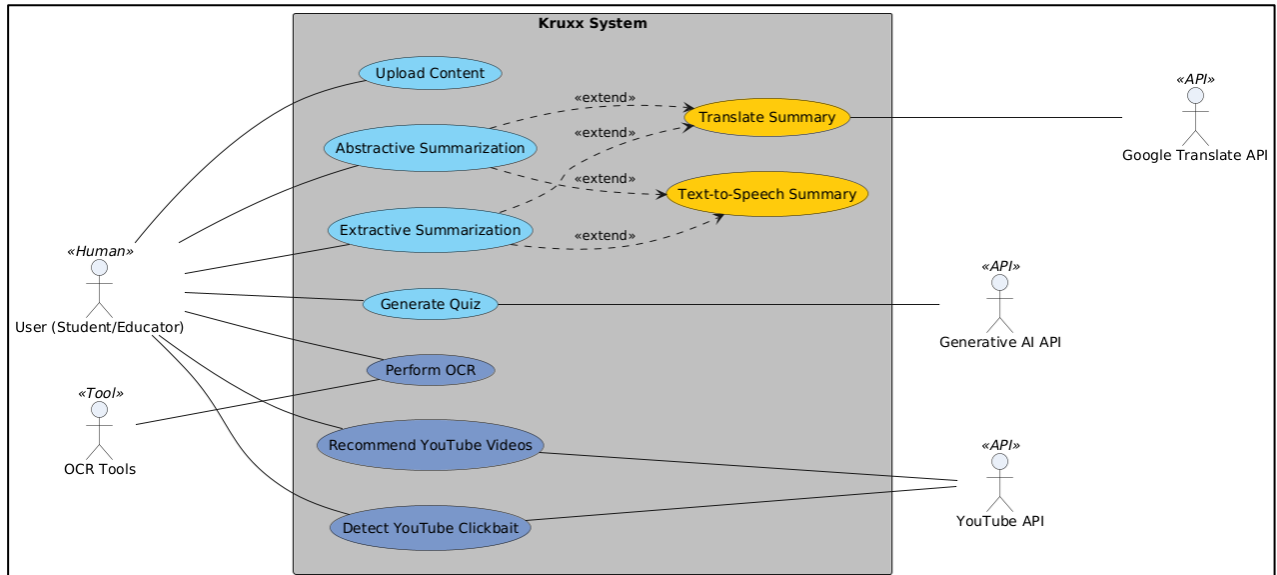


Fig 4.1- Use Case Diagram

The Class Diagram shows all the different functions and functionalities that are in our project and bunches them together in different classes to help show the relationship and process of different classes.

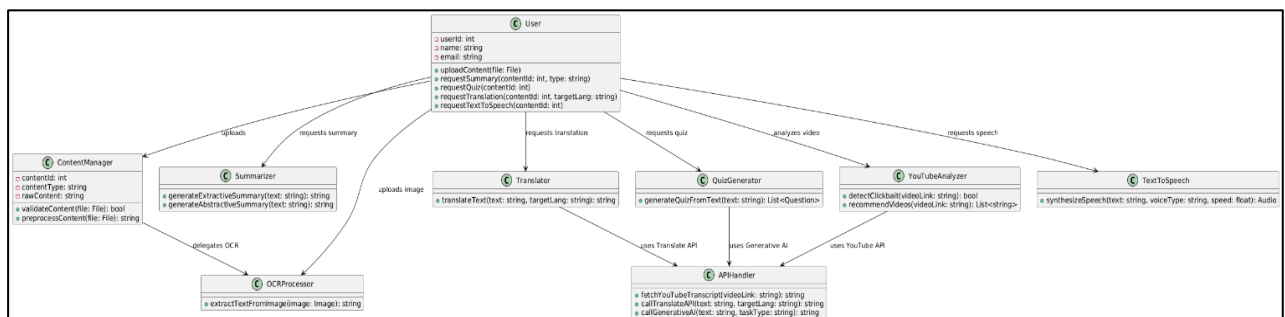


Fig 4.2- Class Diagram

The Control Flow Diagram illustrates the step-by-step execution path of the project when the user interacts with it. It also helps visualize the logical decision-making and flow of control from start to end.

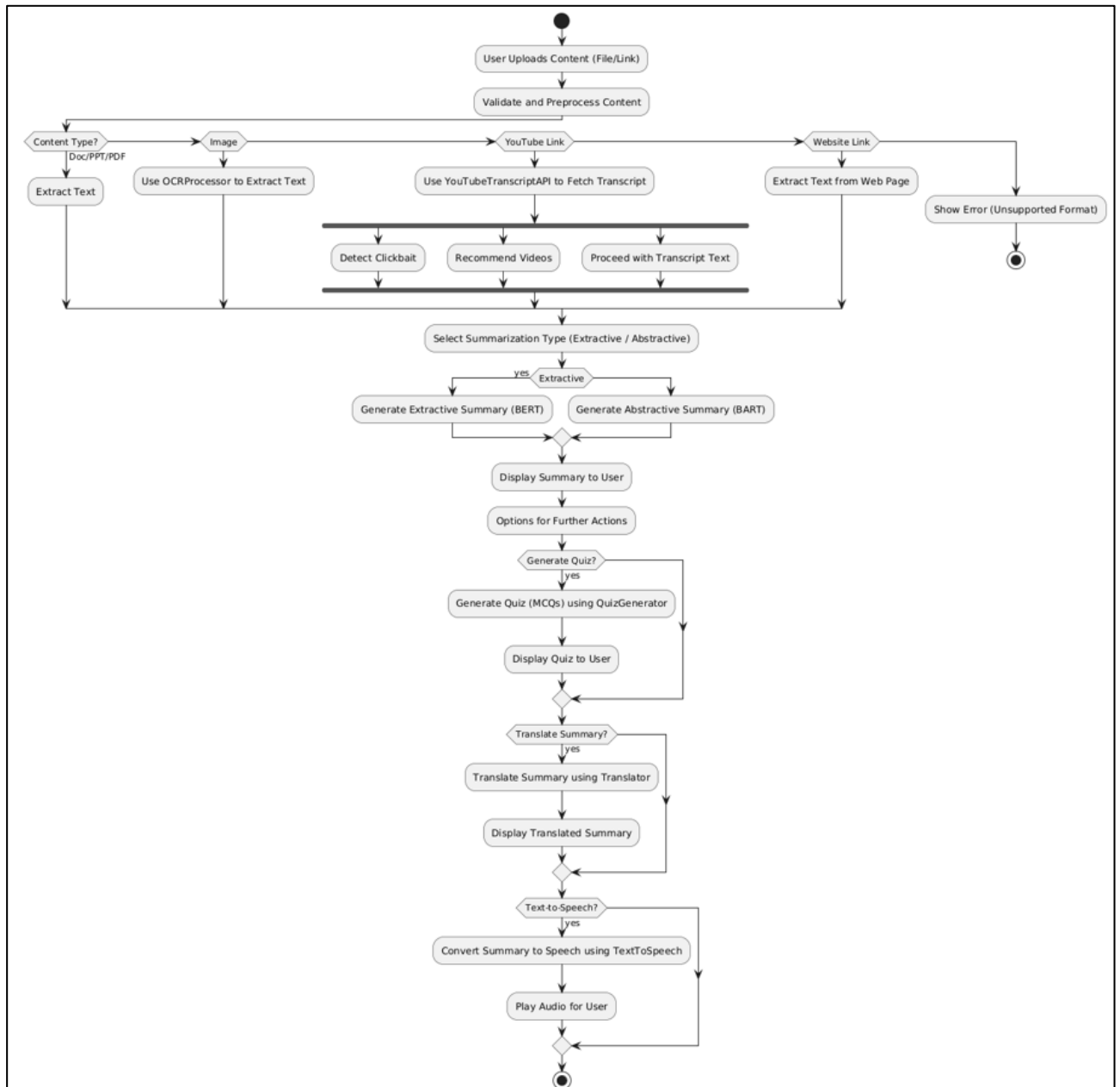


Fig 4.3- Control Flow Diagram

The Sequence Diagram illustrates the dynamic interaction between the user, frontend, backend, and various internal modules during a session. It also captures the time-ordered flow of requests and responses between components in a modular and asynchronous manner.

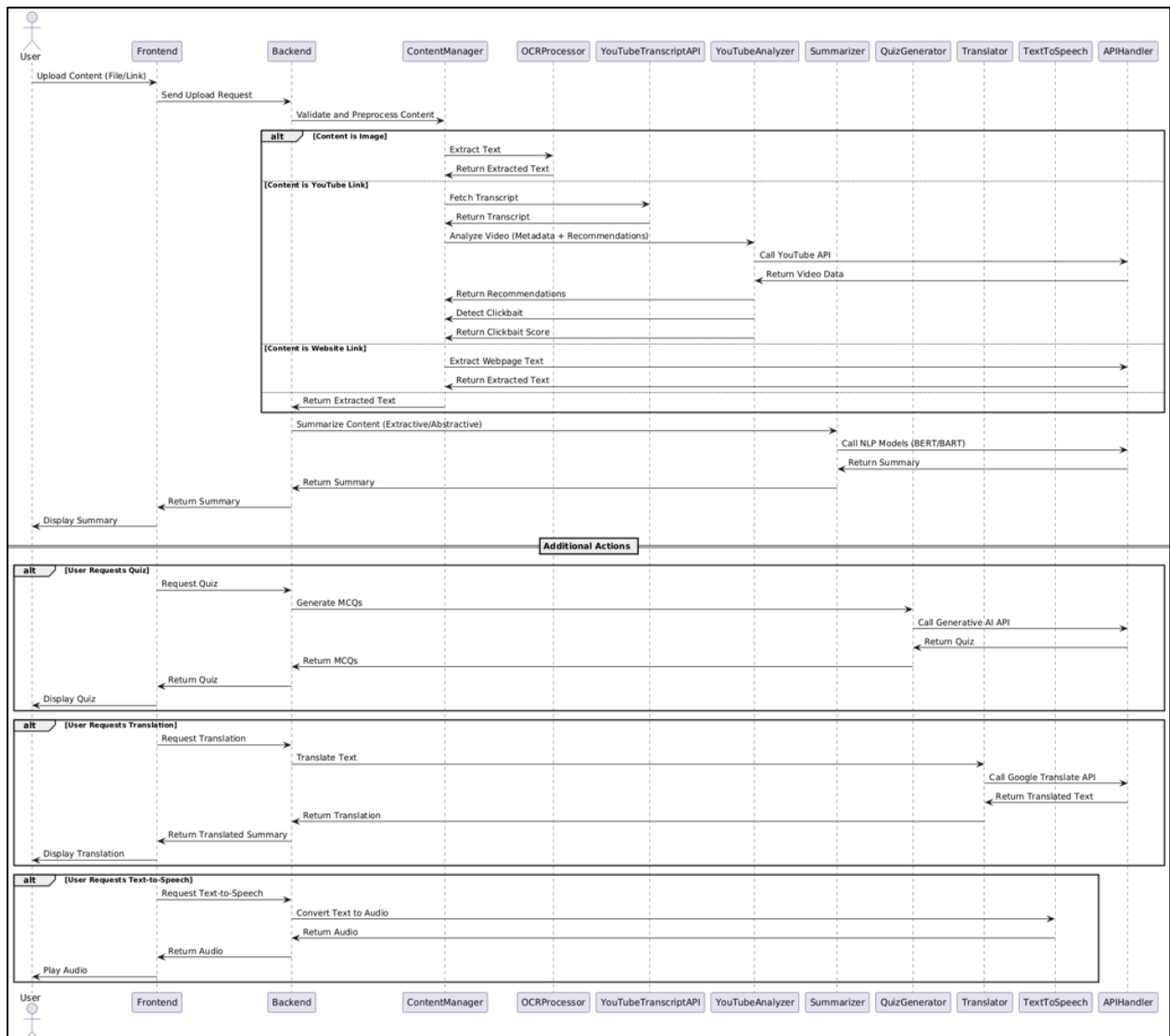


Fig 4.4- Sequence Diagram

4.2) IMPLEMENTATION DETAILS AND ISSUES

1. Backend Development

Activities

1. Set up session management using Flask Sessions.
2. Handle uploaded content and file types, including summaries and quiz data.
3. Integrate NLP models like BERT and BART for efficient summarization.
4. Uses generative AI API to generate quizzes
5. Uses google-translate API for multi-language support
6. Implements algorithms for clickbait YouTube video detection and for generating recommendations.

Technologies

- **Flask:** A lightweight Python web framework designed for API and backend development.
- **Transformers Library:** For working with pre-trained NLP models like BERT and BART.

Deliverables

- Capabilities for uploading, processing, and storing content, summaries, and quiz data.
- Can detect clickbait YouTube videos and generates video recommendations.

2. Summarization and Quiz Generation Modules

Activities

1. **Summarization:**
 - Uses BERT for extractive summarization.
 - Implements BART for abstractive summarization.
2. **Quiz Generation:**
 - Uses generative AI API for quiz generation

Technologies

- **Transformers:** For NLP model integration.

- **Generative AI API:** For quiz generation.

Deliverables

- A fully operational summarization system.
- Dynamic quiz generation.

3. Frontend Development

Activities

1. Design intuitive interfaces for uploading files, selecting summarization options, and taking quizzes.
2. Develop responsive forms that handle various file types (PDF, DOC, images, videos).

Technologies

- **HTML/CSS:** For foundational design and layout.
- **JavaScript:** To add dynamic functionality.
- **React:** For a sleek, responsive UI.

Deliverables

- A responsive frontend that works seamlessly.
- Dynamic pages to showcase summaries and quizzes.

4. Issues Encountered

1. **Model Performance on Large Inputs:** BERT and BART faced difficulties when handling inputs beyond the token limit.
Solution: Split large documents into smaller, manageable chunks.
2. **OCR Accuracy for Handwritten Notes:** The system struggled with unclear handwriting.
Solution: Preprocess images to improve clarity using libraries like PIL.
3. **Complexity of Dynamic Quiz Generation:** Creating distractors that are contextually relevant proved to be tricky.
Solution: Use generative AI API for better distractor options.
4. **Performance Overhead:** Batch processing resulted in high computation times.
Solution: Preload models into memory for faster performance.

4.3) RISK ANALYSIS AND MITIGATION

4.3.1 Risk Identification

Risk ID	Classification	Description of Risk	Risk Area	Probability (P)	Impact (I)	Risk Exposure (P × I)
R1	Development Environment - Usability	Users find it difficult to navigate the platform or use advanced features.	User Experience	Medium (3)	High (4)	12
R2	Product Engineering - Model Accuracy	Summarization models produce inaccurate or irrelevant summaries.	Model Performance	Medium (3)	High (4)	12
R3	Product Engineering - Integration	API failures disrupt communication between frontend, backend.	Backend/API Integration	Medium (3)	High (4)	12
R4	Code and Unit Test - OCR Performance	OCR fails to accurately extract text from low-quality handwritten images.	OCR Reliability	High (4)	Medium (3)	12

4.3.2 Risk Area Analysis

Risk Area	Number of Risk Statements	Weights (In + Out)	Total Weight
User Experience	1	6 + 6	12
Model Performance	1	6 + 6	12
Backend/API Integration	1	3 + 3	6
OCR Reliability	1	6 + 3	9

4.3.3 Mitigation Approaches

Risk ID	Risk Statement	Risk Area	Mitigation Approach	Start Date	Completion Date	Additional Resources Needed
R1	Users find it difficult to navigate the platform or use advanced features.	User Experience	Conduct usability testing and redesign UI based on feedback.	03/04/2025	20/04/2025	Prototyping software

R2	Summarization models produce inaccurate or irrelevant summaries.	Model Performance	Fine-tune models using domain-specific datasets; test summaries rigorously.	01/04/2025	18/04/2025	Fine-tuning datasets, Model evaluation tools
R3	API failures disrupt communication between frontend, backend.	Backend/API Integration	Conduct regular API testing and implement retries for critical endpoints.	08/04/2025	22/04/2025	Integration testing frameworks
R4	OCR fails to accurately extract text from low-quality handwritten images.	OCR Reliability	Preprocess images using filters to enhance quality before OCR.	12/04/2025	25/04/2025	Image preprocessing libraries (PIL)

CHAPTER- 5

TESTING

5.1 Test Plan

Type of Test	Will the test be performed?	Explanation	Software Component
Functional Testing	Yes	Verifies file upload, summarization, quiz generation, clickbait detection and recommendation.	Frontend, Backend, APIs
Unit Testing	Yes	Validates individual modules like summarization and OCR.	Backend APIs, Flask routes
Integration Testing	Yes	Tests interactions between frontend and backend.	Full-stack Integration
Performance Testing	Yes	Validates the speed of NLP processing and queries.	Backend, NLP Models
Stress Testing	No	Simulates high traffic scenarios to check system robustness.	-
Load Testing	Yes	Assesses system performance under typical user workloads.	Full Stack System
Volume Testing	No	Ensures database stability with large datasets.	-
End-to-End Testing	Yes	Validates the entire user journey, from upload to result.	Frontend, Backend
Security Testing	No	Checks for unauthorized access and data encryption.	-
Scalability Testing	No	Tests system behaviour with an increasing user base.	-

5.2 Test Team Details

Role	Name	Responsibilities
Tester	Vansh Gupta	Perform unit, integration, load, testing for reliability.
Developer	Aryan Jolly	Fix issues and optimize backend performance.
Lead QA	Mukund Sarda	Oversee all testing activities, ensure adherence to quality standards.

5.3 Test Schedule

Activity	Start Date	Completion Date	Hours (Approx)
Requirement Gathering	01/04/2025	01/04/2025	1
Obtain Input Data	03/04/2025	04/04/2025	4
Environment Setup	05/04/2025	05/04/2025	2
Unit Testing	07/04/2025	09/04/2025	7
Integration Testing	10/04/2025	11/04/2025	6
Performance Testing	12/04/2025	15/04/2025	8
Load Testing	16/04/2025	17/04/2025	4
User Acceptance Testing	18/04/2025	23/04/2025	6
Bug Fixes and Iterations	24/04/2025	27/04/2025	12

5.4 Test Environment

Category	Details
Software Items	Frontend: React.js, Backend: Flask
Operating Systems	Windows

5.5 Component Decomposition and Type of Testing

S. No	Component	Type of Testing	Technique
1	Frontend (React.js)	Unit, Integration, Load	Black Box
2	Backend (Flask APIs)	Unit, Integration	White Box
4	NLP Models (BERT/BART)	Performance, Accuracy	Tool-Based
5	Quiz Module	Unit, Integration	Black Box

5.6 Test Cases

Test Case ID	Description	Input	Expected Output	Status
T_F1	Validate file upload for supported formats	PDF, Image, DOC	File uploaded successfully.	Pass
T_F2	Handle unsupported file formats	Executable file	Error message: Unsupported file type.	Pass
T_F3	Summarization with BERT	150-word text	Extractive summary generated.	Pass
T_F4	Summarization with BART	150-word text	Abstractive summary generated.	Pass
T_F5	Quiz generation	Text input	10 MCQs with distractors generated.	Pass
T_BE1	API response under high traffic	Concurrent requests	Responses within acceptable thresholds	Pass
T_D2	Validate large file upload	50MB PDF	File processed and stored successfully.	Pass
T_F6	Handle quiz submission	Quiz answers	Correct and incorrect answers highlighted.	Pass

T_F7	Clickbait detection	YouTube link	Yes/No depending on link	Pass
T_F8	Video Recommendation	YouTube link	Recommend videos based on inputted link	Pass
T_F9	Translate	Summarized result	Translated summary	Pass
T_F10	Text-to-Speech	Translated summary	Audible version of translated summary	Pass

5.7 Error and Exception Handling

Test Case ID	Test Case Description	Debugging Technique
T_F3	Summarization with incorrect output	Fine-tuned NLP model
T_BE1	API crashes uner high traffic	Middleware optimization
T_F6	Incorrect quiz results	Refactored question generation logic

5.8) LIMITATIONS OF THE SOLUTION

5.8.1. Technical Limitations

5.8.1.1 Dependency on Pre-Trained Models

- Description:** The summarization and quiz generation functionalities rely extensively on pre-trained NLP models such as BERT and BART. While these models are highly advanced and have transformed natural language processing in many ways, they do come with certain inherent limitations. One such limitation is the Input Token Limit. Both BERT and BART are restricted to a maximum of 512 tokens, which makes it difficult to process larger documents without first splitting them into smaller, more manageable chunks. Another issue is Generalization. These models, although powerful, may not perform at their best when faced with domain-specific or highly technical content such as medical or legal texts, unless they are subjected to further fine-tuning.
- Impact:** There is a reduction in the accuracy and relevance of the summaries and quiz questions when dealing with niche subject areas. Additionally, the need to split large

documents into smaller chunks and then recombine the outputs leads to an increase in overall processing time.

5.8.1.2 OCR Performance

- **Description:** The system makes use of OCR tools such as EasyOCR and DocTR to extract text from images or scanned documents. However, this process is not without its challenges. Low-resolution or poor-quality images often result in inaccurate text extraction, thereby affecting the overall output. Furthermore, handwritten notes, particularly those written in cursive, are difficult to process with a high degree of precision.
- **Impact:** Errors in the extracted text can lead to inaccuracies in the generated summaries and result in quiz questions that are irrelevant or misaligned with the original content.

5.8.1.3 High Computational Requirements

- **Description:** The processes of summarization and quiz generation involve computationally intensive tasks, especially when deep learning models are employed. This, in turn, can lead to certain challenges such as increased latency when handling large documents or multiple concurrent user requests.
- **Impact:** Reduced user experience due to delays in processing.

5.8.2. Operational Limitations

5.8.2.1 Limited Customization for Users

- **Description:** Although the platform offers options for selecting the summary type (extractive/abstractive) and length (short, medium, or long), there is limited scope for deeper customization.
- **Impact:** Users may find themselves unable to specify additional preferences, such as focusing on specific sections of content or customizing the quiz formats to suit their needs.

5.8.2.2 Dependency on Internet Connectivity

- **Description:** The system is built as a web application, which means that stable internet connectivity is required to access services such as summarization and quiz generation.
- **Impact:** Users in areas with limited or unreliable internet access may encounter significant difficulties in fully utilizing the platform.

5.8.2.3 Manual Intervention for Edge Cases

- **Description:** In some instances, particularly when dealing with extremely noisy images or overly complex documents, manual intervention may be necessary to refine either the inputs or the outputs to ensure accuracy.
- **Impact:** Reduces the level of automation promised by the system.

5.8.3. User Experience Limitations

5.8.3.1 Learning Curve

- **Description:** While the platform is designed to be user-friendly, first-time users may find certain features, such as selecting the summary length, to be somewhat overwhelming.
- **Impact:** Could lead to a reduced initial adoption rate or lower user satisfaction.
- **Mitigation:** To enhance the user experience, onboarding tutorials, tooltips, and a guided walkthrough could be introduced.

5.8.4. Quiz Generation Limitations

5.8.4.1 Lack of Customization

- **Description:** Currently, the quiz only generates Multiple Choice Questions (MCQs) and gives 10 questions with 4 options each. There are no customizable options to change the type of question or the number of them.
- **Impact:** Could reduce the effectiveness of quizzes as a tool for assessment.
- **Mitigation:** Could implement a window of options to customize the quiz as per user needs.

5.8.4.2 Lack of Adaptive Quizzes

- **Description:** The platform does not yet provide quizzes that adapt dynamically based on user performance, i.e., it doesn't fine-tune itself to ask more about the topics where the user might be failing at.
- **Impact:** Missed opportunities to tailor quizzes to individual learning needs.
- **Mitigation:** Incorporating adaptive learning algorithms could enhance the personalization of quizzes.

CHAPTER- 6

FINDINGS, CONCLUSION, AND FUTURE WORK

6.1) FINDINGS

Kruxx has proven to deliver significant value in streamlining the exam preparation process for both students and educators. Key findings include:

1. Efficiency:

- Automated summarization and quiz generation effectively reduce the time and effort needed for studying and preparing teaching materials.
- Customization options, such as summary length and type, are designed to accommodate a variety of learning preferences.

2. User Feedback:

- Initial testing and user feedback suggest high satisfaction with the platform, primarily due to its user-friendly interface and comprehensive feature set.
- Users also appreciate the ability to support multimedia inputs, such as images and videos, which enhances the platform's versatility.

3. Challenges Addressed:

- OCR accuracy issues, particularly for low-quality handwritten notes, were successfully tackled by integrating advanced preprocessing techniques.
- NLP models were optimized for handling large documents through effective chunking and batching methods, improving overall performance.

4. Competitive Advantage:

- In comparison to existing solutions, **Kruxx** offers a unique and integrated combination of summarization, quiz generation, clickbait YouTube video detection as well as YouTube video recommendation, setting it apart as a comprehensive platform for not only exam preparation but learning in general for students and teachers alike.

6.2) CONCLUSION

Kruxx successfully meets its goal of delivering a robust and user-friendly solution for summarizing study materials and generating personalized quizzes. By utilizing advanced technologies such as BERT and BART, the platform efficiently addresses key challenges in learning like information overload and time constraints. Its capacity to handle diverse input formats and even generate quizzes based on the given input distinguishes it from other existing solutions.

Key achievements include:

- Automated summarization and quiz generation.
- Support for multimedia inputs.
- Clickbait YouTube video detection and smart video recommendation generation.
- High user satisfaction rates during testing phases.

Kruxx is an ideal solution for students, educators as well as institutions aiming to optimize the learning process and enhance educational outcomes.

6.3) FUTURE WORK

While the current version of **Kruxx** is robust and feature-rich, there are opportunities for further enhancement:

1. Adaptive Learning

- Integrate AI-based adaptive learning techniques to personalize quizzes and feedback, adjusting to each user's performance and learning pace.

2. Real-Time Collaboration

- Enable real-time collaboration features to facilitate group study sessions or educator-student interactions, enhancing the overall learning experience.

3. Advanced Analytics

- Incorporate predictive analytics to identify learning patterns and suggest customized study schedules or resources, ensuring more effective learning pathways.

4. Mobile App Development

- Develop dedicated mobile applications for Android and iOS platforms, improving accessibility and convenience for users on the go.

5. Institutional Integration

- Provide API integrations with popular learning management systems (LMS) such as Moodle, Blackboard, and Google Classroom, fostering broader adoption within educational institutions.

6. Gamification

- Introduce gamification elements like badges, leaderboards, and rewards to encourage user engagement and make the learning experience more interactive.

7. Cloud Deployment

- Deploy a cloud-based architecture (e.g., AWS, Azure) to improve scalability, performance, and overall system reliability.

REFERENCES

- [1] McGraw Hill, "McGraw Hill Study Trends Report: Instructors and Students on Study Habits and Technology Use," McGraw Hill News & Insights, Oct. 2023. [Online]. Available: <https://www.mheducation.com/news-insights/blog/mcgraw-hill-study-trends-report.html>
- [2] D. Lovin and D. Bernardeau-Moreau, "Stress among Students and Difficulty with Time Management: A Study at the University of Galati in Romania," *Social Sciences*, vol. 11, no. 12, p. 538, Nov. 2022, Doi: 10.3390/socsci11120538.
- [3] McGraw Hill, "McGraw Hill Study Trends Report: Instructors and Students on Study Habits and Technology Use," McGraw Hill News & Insights, Oct. 2023. [Online]. Available: <https://www.mheducation.com/news-insights/blog/mcgraw-hill-study-trends-report.html>
- [4] Learnosity, "A Third of US Teachers Considered Leaving Education Due to Grading Workload, Says New Research by Learnosity," *Business Wire*, Mar. 26, 2025. [Online]. Available: <https://www.businesswire.com/news/home/20250326730498/en/A-Third-of-US-Teachers-Considered-Leaving-Education-Due-to-Grading-Workload-Says-New-Research-by-Learnosity>
- [5] Varshney, D., Vishwakarma, D.K. A unified approach for detection of Clickbait videos on YouTube using cognitive evidences. *Appl Intell* 51, 4214–4235 (2021). [Online]. Available: <https://doi.org/10.1007/s10489-020-02057-9>
- [6] M. Lewis et al., "BART: Denoising Sequence-to-Sequence Pre-training for natural language generation, Translation, and Comprehension," arXiv.org, Oct. 29, 2019. [Online]. Available: <https://arxiv.org/abs/1910.13461>
- [7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv.org, Oct. 11, 2018. [Online]. Available: <https://arxiv.org/abs/1810.04805>
- [8] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization," arXiv.org, Dec. 18, 2019. [Online]. Available: <https://arxiv.org/abs/1912.08777>
- [9] J. Baek et al., "What is wrong with scene text recognition model comparisons? dataset and model analysis," arXiv.org, Apr. 03, 2019. [Online]. Available: <https://arxiv.org/abs/1904.01906>

- [10] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A Critical Review of Recurrent Neural Networks for sequence Learning,” arXiv.org, May 29, 2015. [Online]. Available: <https://arxiv.org/abs/1506.00019>
- [11] A. Vaswani et al., “Attention is all you need,” arXiv.org, Jun. 12, 2017. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [12] V. Raina and M. Gales, “Multiple-Choice question generation: towards an automated assessment framework,” arXiv.org, Sep. 23, 2022. [Online]. Available: <https://arxiv.org/abs/2209.11830>

21103105 Kruxx

ORIGINALITY REPORT

6%

SIMILARITY INDEX

4%

INTERNET SOURCES

2%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1

www.slideshare.net

Internet Source

2%

2

www.coursehero.com

Internet Source

<1%

3

Submitted to Keyano College

Student Paper

<1%

4

skagitcounty.net

Internet Source

<1%

5

github.com

Internet Source

<1%

6

webthesis.biblio.polito.it

Internet Source

<1%

7

Submitted to Fakultet elektrotehnike i računarstva / Faculty of Electrical Engineering and Computing

Student Paper

<1%

8

Submitted to University of East London

Student Paper

<1%

9

core.ac.uk

Internet Source

<1%

10

export.arxiv.org

Internet Source

<1%

11

Submitted to Indian Institute of Technology Guwahati

Student Paper

<1%

12	scientific-publications.ukaea.uk Internet Source	<1 %
13	web.archive.org Internet Source	<1 %
14	Submitted to University of Florida Student Paper	<1 %
15	Submitted to Harrisburg University of Science and Technology Student Paper	<1 %
16	Submitted to University of Mindanao Student Paper	<1 %
17	Submitted to University of Wollongong Student Paper	<1 %
18	Submitted to University of Westminster Student Paper	<1 %
19	onlinelibrary.wiley.com Internet Source	<1 %
20	Submitted to Arab Open University Student Paper	<1 %

Exclude quotes On
Exclude bibliography On

Exclude matches < 14 words