LLM BASED EDUCATIONAL FRAMEWORK FOR TEACHERS AND STUDENTS

A PROJECT REPORT

Submitted by

GOGILARASAN S 2021115036

GOKULRAJ S 2021115038

VIKHRAM S S 2021115119

SAIGANESH V 2021115316

submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment for the award of the degree
of

BACHELOR OF TECHNOLOGY
in

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600 025

APRIL 2025

ANNA UNIVERSITY

CHENNAI - 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled **LLM BASED EDUCATIONAL FRAMEWORK FOR TEACHERS AND STUDENTS** is the bonafide work of GOGILARASAN S (2021115036), GOKULRAJ S (2021115038), VIKHRAM S S (2021115119), SAIGANESH V (2021115316) who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE: Chennai	**Dr. S BAMA**
DATE:	**ASSOCIATE PROFESSOR**
	PROJECT GUIDE
	DEPARTMENT OF IST, CEG
	ANNA UNIVERSITY
	CHENNAI 600025

COUNTERSIGNED

Dr. S SWAMYNATHAN
PROFESSOR AND HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

Empowering accessible learning for students with disabilities is crucial for creating an inclusive educational environment. This project focuses on developing automated assistive educational applications that enable educators to efficiently generate and customize tools for students supporting Augmentative and Alternative Communication. This system integrates various resources, such as interactive games and educational quiz templates, into a unified user interface hub, streamlining the teaching process while enhancing accessibility for students through a single-click interface.

A Python-based backend integrates key functionalities and APIs, including Large Language Models (LLMs), while a React, Vite, and CSS-powered frontend ensures a seamless user experience. This architecture facilitates the generation of templates for quizzes, games, and language learning activities optimized for single-click interactions, catering specifically to students with limited mobility. By utilizing APIs and enabling dynamic UI customization, each template is designed to be adaptable, visually engaging, and easily navigable, addressing the unique learning needs and styles of students.

This system provides a scalable solution for educational institutions aiming to support students with special needs. By automating the generation of assistive applications, barriers for educators are reduced, creating a more inclusive and personalized learning experience. Features such as single-click operations, text-to-speech, and on-screen navigation tools create a vision for a future where education is accessible to all.

ACKNOWLEDGEMENT

We wish to record our deep sense of gratitude and profound thanks to our project guide **Dr. S. Bama**, Associate Professor, Department of Information Science and Technology, College of Engineering, Guindy for her keen interest, inspiring guidance, constant encouragement with our work during all stages, to bring this project into fruition.

We would like to convey our gratitude to **Dr. S. Swamynathan**, Professor & Head, Department of Information Science and Technology, Anna University, Chennai for providing us with the opportunity and infrastructure to carry out this project. We would also like to express our sincere thanks to the panel of reviewers **Dr. K. Vani**, Professor, **Dr. K. Indra Gandhi**, Associate Professor, **Dr. K. Vidya**, Associate Professor, **N.Anbarasi**, Teaching Fellow, Department of Information Science and Technology for their valuable suggestions throughout the course of our project.

We thank our parents, family, and friends for bearing with us during the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

GOGILARASAN S
GOKULRAJ S
VIKHRAM S S
SAIGANESH V

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AA	Accessibility Assurance
ADA	Americans with Disabilities Act
API	Application Programming Interface
ARIA	Accessible Rich Internet Applications
CRUD	Create, Read, Update, Delete (database operations)
CSS	Cascading Style Sheets
GPT	Generative Pre-trained Transformer
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
LLM	Large Language Model
MCQ	Multiple Choice Questions
MVC	Model-View-Controller
OOP	Object-Oriented Programming
SQL	Structured Query Language
TK	Tkinter
TTS	Text-to-Speech
UI	User Interface
UX	User Experience
WCAG	Web Content Accessibility Guidelines

CHAPTER 1

INTRODUCTION

In recent years, the demand for accessible and inclusive learning tools has significantly increased, especially for students with disabilities who face unique challenges in traditional educational settings. This project focuses on developing an interactive quiz and educational game application that provides teachers with customizable templates. These templates enable teachers to create engaging learning experiences that are tailored to the needs of students, making learning more accessible. The primary focus is to create applications that are easy to navigate with minimal physical input, enabling disabled students to participate actively in their education with minimal assistance. Our approach combines artificial intelligence and custom templates to generate and customize applications that fit individual needs, and target accessibility features. By automating application generation, we simplify the process for educators, reducing the technical barrier to create tailored learning tools. Additionally, the platform includes an AI-driven Education Assistant to help teachers enhance content creation and customization, utilizing a Retrieval-Augmented Generation system to further support educators in delivering targeted, accessible learning experiences This project offers a structured hub containing various learning applications spanning quizzes, language exercises, games, and other educational tools each designed with a single-click interface,making the learning process more intuitive for those with mobility challenges.

1.1 ASSISTIVE TECHNOLOGY IN EDUCATION

Assistive technology has transformative applications across diverse domains, each aimed at improving accessibility, usability, and inclusivity. These applications empower individuals by providing tailored tools, enabling them to interact with their environment and perform tasks independently.

Education is one of the primary domains where assistive technologies create significant impact (Figure 1.1). These technologies bridge learning gaps for students with disabilities, promoting inclusivity and accommodating diverse learning needs.

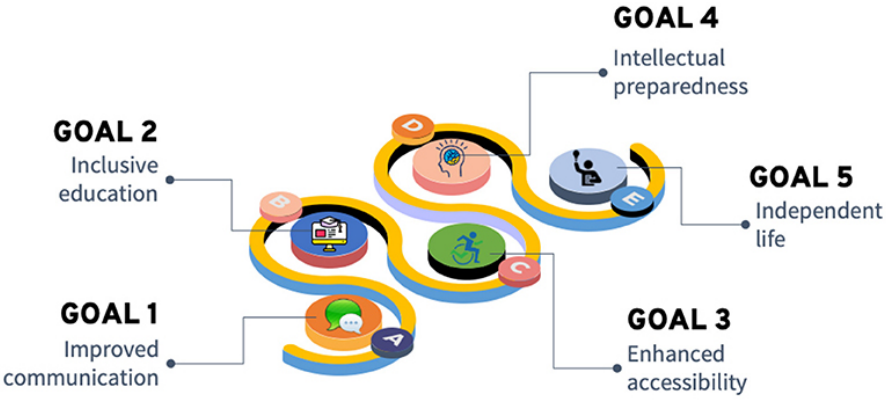

Figure 1.1: Cutting-edge communication and learning assistive technologies for differently-abled children

- **Specialized Learning Modules:** Assistive technologies provide learning modules designed for students with cognitive or physical challenges. Tools like on-screen keyboards or voice-controlled navigation systems help students with motor difficulties participate in learning activities effectively.

- **Interactive and Game-Based Learning:** Gamification elements, such as quizzes, flashcards, and storytelling applications, make learning engaging and enjoyable, motivating students to stay focused while improving retention.

- **Accessible Course Material Distribution:** Text-heavy educational materials can be converted into audio, visual, or interactive formats, enabling students with visual or auditory impairments to access content on par with their peers.

- **Remote Learning Support:** With the growth of online education, assistive technologies ensure that students with disabilities can seamlessly participate in virtual classrooms and complete assignments independently.

1.2 BACKGROUND

The importance of accessible learning solutions has become increasingly evident as the world shifts toward digital education. Traditional educational software and tools often lack the flexibility needed to accommodate diverse accessibility requirements, especially for students with severe physical impairments. Recognizing these limitations, researchers and developers are now focusing on creating tools that cater specifically to individuals who rely on minimal interaction methods, such as single-click navigation.

Large Language Models offer a unique opportunity to facilitate the rapid generation of customizable, accessible applications without requiring extensive programming skills. By employing LLMs and assistive templates, educators and caregivers can quickly create applications that provide tailored support for students with disabilities. This project integrates these LLM-driven capabilities with a centralized UI hub, where various assistive tools, including on-screen keyboards and basic learning games, can be accessed coherently.

1.3 CHALLENGES

Despite the advances in educational technology and assistive systems, significant challenges remain that hinder their widespread adoption and efficacy. This section discusses existing challenges, highlights research gaps, and underscores the necessity of the proposed system.

1.3.1 Existing Challenges

Educational tools and assistive technologies face several limitations that affect their usability and inclusiveness:

- **Limited Personalization:** Many existing systems lack the ability to dynamically adapt content to individual user needs. For example, content difficulty or accessibility features are often rigid, failing to accommodate the diverse requirements of students with disabilities.

- **High Development Costs:** Custom solutions typically demand significant technical expertise and financial resources, making them inaccessible for smaller institutions or educators.
- **Lack of Engagement:** Traditional learning methods for students with disabilities tend to be monotonous, lacking interactive or gamified elements to sustain user interest.
- **Insufficient Accessibility Standards:** A large number of educational tools do not adhere to accessibility guidelines like WCAG, limiting their usability for individuals with visual, auditory, or motor impairments.
- **Dependency on Manual Efforts:** The reliance on educators or developers to manually create content or applications results in slower deployment and increased chances of inconsistency in quality.

1.3.2 Knowledge Gaps Addressed by the Proposed System

The proposed system bridges these gaps by integrating modern technologies and innovative methodologies:

- **LLM-Based Content Generation:** The system leverages Large Language Models (LLMs) to generate content dynamically, allowing the creation of engaging and tailored learning experiences based on user inputs such as topic and difficulty level.
- **No-Code Development Platform:** By enabling educators to create applications without coding knowledge, the system democratizes the

creation of assistive applications, making it accessible to non-technical users.

- **Advanced Customization Options:** Features such as dynamic color selection, font size adjustments, and text-to-speech features ensure that the applications meet diverse accessibility requirements.
- **Integrated Accessibility Features:** Tools like on-screen keyboards, single-click navigation, and voice assistance enhance usability for students with limited mobility or other impairments.
- **Interactive and Engaging Learning Modules:** Incorporating gamification and interactivity sustains user engagement and ensures a more effective learning experience.

1.3.3 Comparison of Challenges and Features

The following table outlines how the proposed system addresses existing challenges:

Table 1.1: Comparison of Existing Challenges and Proposed Features

Challenges in Existing Systems	**Features of the Proposed System**
Limited personalization of content	Dynamic content generation based on user inputs (e.g., topic, difficulty level) using LLMs
High development costs	No-code platform enabling non-technical users to create applications
Lack of engaging and interactive features	Gamified modules, interactive quizzes, and visually appealing templates
Insufficient adherence to accessibility standards	Comprehensive accessibility options, including voice prompts, on-screen navigation, and customizable UIs
Dependency on manual efforts for application creation	Automation of content generation and template customization

1.3.4 Need and Motivation for the Proposed System

The proposed system emerges as a solution to address critical gaps in existing educational tools, particularly for students with disabilities. Current systems often lack the flexibility and inclusivity required to cater to diverse needs, leading to limited engagement and effectiveness. By employing modern technologies like Large Language Models, no-code platforms, and advanced customization options, this system empowers educators to create dynamic, accessible, and tailored learning experiences with ease.

This innovative approach fosters independence and inclusivity for students with disabilities by integrating features such as TTS features, single-click navigation, and customizable interfaces. Educators are equipped with user-friendly tools to design applications without requiring extensive technical expertise, breaking barriers to accessibility in education. The proposed system not only bridges the technological gaps but also paves the way for the next generation of accessible educational solutions, aligning with global efforts to create equitable learning environments for all.

1.4 OBJECTIVES

The primary objective of this project is to develop a comprehensive platform that enables the creation of customizable, accessible learning applications for students with disabilities. This platform should:

- Empower educators and caregivers to generate no code-applications that cater to specific accessibility needs without requiring programming expertise.
- Provide a range of learning templates, including quizzes, games, and language tools, that can be customized for different levels and topics.

- Ensure single-click navigation across all applications, allowing paralyzed users to interact with the content smoothly.
- Offer dynamic features such as color selection, text-to-speech features, and automatic traversal to enhance user engagement and satisfaction.

1.5 PROBLEM STATEMENT

Students with Augmentative and Alternative Communication needs, including those with limited mobility or speech impairments, often face significant challenges in accessing traditional educational tools. Many standard applications require complex interactions or extensive physical movement, making it difficult for these students to navigate content independently. Additionally, creating accessible, customized learning applications that cater to individual needs often requires technical knowledge, which can be a barrier for educators without programming skills. Given these challenges, there is a pressing need for a solution that provides accessible, easy-to-use learning tools specifically designed for students with AAC needs. This project addresses that gap by developing a platform that simplifies the creation of customizable, accessible learning tools, incorporating features like TTS images, sounds, etc. The platform enhances user interaction through single-click operations, ensuring students can engage with content more independently and effectively.

1.6 PROPOSED SOLUTION

To address these challenges, this project proposes an AI-powered educational platform designed for students with disabilities. It simplifies the creation of accessible learning tools by providing teachers with customizable templates for interactive quizzes, educational games, and other content, all without requiring technical expertise. Key features of the platform include:

1. **Customizable Templates**

   - Teachers can create different types of interactive quizzes, such as Multiple Choice Questions (MCQs), Match the Following, Sequence Finder, engaging conversation builder, etc.
   - Each template is fully customizable, allowing teachers to adjust visual settings (scan speed, font color, background color) and content to match the needs of individual students.

2. **Single-Click Interactions**

   - The platform supports single-click operations, making it easier for students with limited mobility to navigate and interact with educational content.

3. **Text-to-Speech (TTS) and Visuals**

   - The TTS feature reads aloud the content, supporting students with reading difficulties or visual impairments.
   - Visual elements, such as images and icons, are integrated into quizzes and games, enhancing comprehension and engagement for students with varying learning styles.

4. **AI-Driven Content Creation**

   - The system uses Retrieval-Augmented Generation (RAG) to assist teachers in generating context-based questions and educational content.
   - The AI-powered content is dynamically created and tailored to suit the specific learning objectives and needs of students.

5. **Interactive Educational Games**

   - Games like Ninja Math Battle, Word Builder, and Sequence Builder are designed to make learning engaging while promoting skills such as math, vocabulary, and logical sequencing.

- Teachers can customize these games by adjusting features like difficulty level, game speed, and content, making them adaptable to the unique needs of each student.

By integrating these features, the platform provides an intuitive, all-in-one solution that empowers educators to create tailored, accessible learning experiences without the need for technical skills. The goal of the platform is to foster an inclusive learning environment where all students, particularly those with AAC needs, can actively participate and thrive in their education.

1.7 ORGANIZATION OF THE THESIS

- **Chapter 1: Introduction** - Provides background, objectives, problem statement, solution overview, and report structure.
- **Chapter 2: Literature Survey** - Reviews existing research on code generation, no-code platforms, assistive technologies, and LLM based applications.
- **Chapter 3: System Design of Application Generation** - Describes system architecture, template source, LLM integration, UI customization, and accessibility features.
- **Chapter 4: Implementation and Results** - Details the development of the platform, UI, algorithms, integration with LLMs, testing and performance analysis
- **Chapter 5: Conclusion and Future Work** - Summarizes findings and suggests future developments for improving the system.

CHAPTER 2

LITERATURE SURVEY

This chapter reviews previous research relevant to accessible learning applications, template-based code generation, and assistive technologies. By analyzing the limitations of existing studies and solutions, we identify areas for improvement that our project aims to address. The section is organized to cover various domains, including template-based code generators, no-code platforms, and assistive technology tools designed for individuals with disabilities.

2.1 TEMPLATE-BASED CODE GENERATORS

Template-based code generation has gained significant traction in software development, particularly for streamlining repetitive tasks and improving development efficiency. Uyanik, B. and Şahin, V. H. (2020) [1] introduced a notable example of this approach with a template-based code generator tailored for Enterprise Resource Planning (ERP) systems. Their work demonstrated a profound impact on development speed, achieving a remarkable 98.95% reduction in average development time compared to manual coding. Integrated into the TBYS ERP system, the generator automated the creation of robust front-end interfaces, back-end logic, database operations, and web services. By adhering to predefined templates and coding practices, the system ensured consistency, error reduction, and improved maintenance.

While this generator proved its scalability and reliability within the ERP domain, it also highlighted certain limitations. For instance, the predefined templates restricted customization options, making the system less flexible for applications requiring unique designs. Additionally, its focus on the Model-View-Controller (MVC) architecture limited adaptability to other architectural patterns, and its implementation using C# and ASP.NET MVC made it domain-specific. Developers unfamiliar with T4 and Handlebars templating tools faced a steep learning curve, and the generator's dependency on robust reference implementations meant that errors in templates or inputs could propagate throughout the system. These challenges underscore the need for broader applicability and flexibility in template-based code generation systems.

Our approach, proposed during this literature survey, intends to address these limitations by extending the scope of template-based generators to create accessible learning applications specifically for students with disabilities. By focusing on accessibility, this concept aims to bridge the gap between enterprise systems and the needs of inclusive education. Unlike traditional systems, the proposed system tailors templates to incorporate features such as customizable user interfaces, voice-based navigation, and compatibility with assistive devices. This focus on accessibility not only enhances the utility of template-based code generators but also demonstrates their potential beyond traditional domains like ERP systems. The envisioned work aspires to provide a versatile and inclusive solution that empowers both educators and students through technology.

2.2 NO-CODE DEVELOPMENT PLATFORMS FOR MULTI-AGENT SYSTEMS

No-code platforms have emerged as transformative tools for enabling application development without traditional programming, and their applicability has expanded to complex domains, including multi-agent systems. Dibia et al. (2024) [2] introduced AUTOGEN STUDIO, a pioneering no-code platform designed for creating and debugging multi-agent workflows. This platform provides a user-friendly drag-and-drop interface, simplifying the configuration of agents, tools, and workflows. Its reusable template gallery accelerates development processes, while profiling features such as message tracking and performance visualizations enhance debugging capabilities. By integrating with backend APIs, AUTOGEN STUDIO facilitates seamless deployment and export of workflows, lowering the barrier to entry for developers and promoting modular approaches in multi-agent system design.

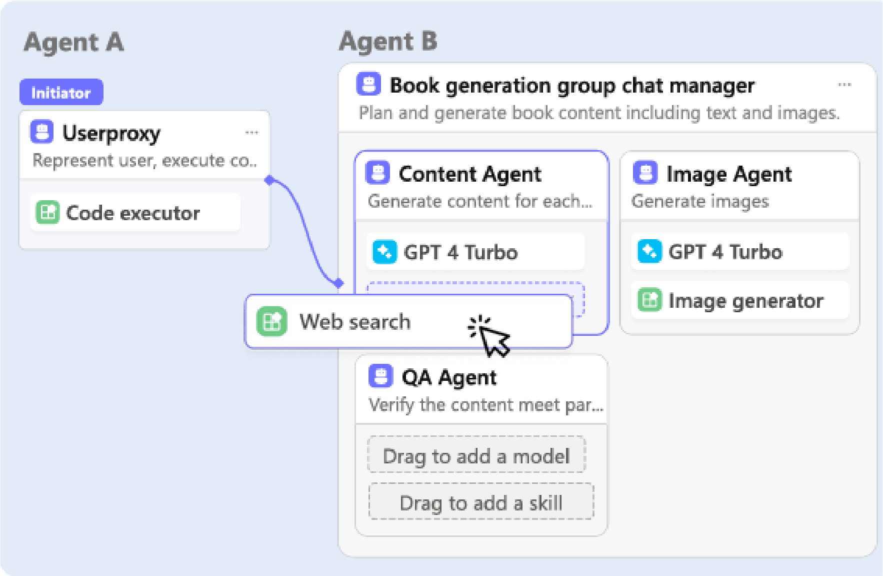
Agent A
Initiator
Userproxy
Represent user, execute co..
Code executor
Agent B
Book generation group chat manager
Plan and generate book content including text and images.
Content Agent
Generate content for each...
GPT 4 Turbo
Web search
Image Agent
Generate images
GPT 4 Turbo
Image generator
QA Agent
Verify the content meet par...
Drag to add a model
Drag to add a skill

Figure 2.1: AUTOGEN STUDIO Interface for Multi-Agent Workflow Composition.

However, while AUTOGEN STUDIO excels in developer-centric scenarios, it does not address the specific needs of end-users with disabilities or applications focused on accessibility and educational content. Its reliance on JSON-based specifications and limited real-time adaptive capabilities make it challenging for non-technical users to configure and manage agent workflows effectively shown in the Figure 2.1. Additionally, the platform lacks collaborative development features and requires additional tooling for production deployment, highlighting gaps in its scalability and adaptability for domain-specific applications.

Our project builds upon the concepts demonstrated by platforms like AUTOGEN STUDIO but shifts the focus to empowering educators. By creating a no-code platform tailored for designing accessible learning applications, we aim to prioritize inclusivity and simplicity, enabling educators without technical expertise to develop tools that cater to the unique needs of students with disabilities. This approach bridges the gap between innovative no-code development and the urgent demand for accessibility-focused educational solutions.

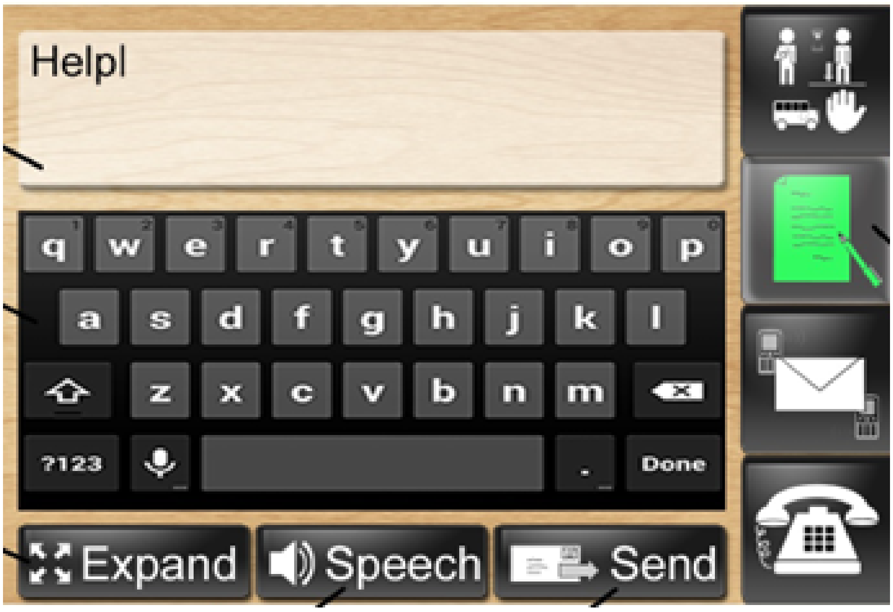

Figure 2.2: Assistive User Interface Design for Mobile Applications.

2.3 ASSISTIVE TECHNOLOGY FOR PEOPLE WITH DISABILITIES

Assistive technology plays a crucial role in enabling individuals with disabilities to access digital content and interact with applications effectively. It focuses on bridging the gap between traditional interfaces and the unique needs of users with physical, cognitive, or developmental impairments. The development of accessible user interfaces for this demographic has been an active area of research, with advancements often centered on improving usability and accessibility. One notable contribution is by Hameed, Ali, and Ali (2022) [3], who developed an assistive computing system utilizing laser-based control and LDR sensor-equipped keyboards for users unable to operate conventional input devices. This system eliminates the need for physical keyboards and mice, marking a significant step forward in hardware-based assistive technology. However, challenges such as power efficiency and the usability of the system for a broader audience remain unresolved, limiting its widespread adoption.

Figure 2.2 illustrates a conceptual design of an assistive user interface for mobile applications. While it underscores the importance of accessible navigation and interaction, its reliance on hardware solutions limits scalability across diverse use cases. Our project extends this idea by adopting a software-driven approach, focusing on customizable templates that eliminate the dependency on conventional input mechanisms. This shift from hardware to software enables the creation of digital learning tools that are more adaptable and accessible, particularly for students with disabilities. By addressing limitations such as flexibility and user autonomy, our solution offers a comprehensive alternative to existing systems.

Assistive technology for individuals with Intellectual and Developmental Disabilities (IDD) has also been explored by Borblik et al. (2015) [4], who emphasized the importance of designing simplified user interfaces for mobile applications. Their approach prioritized navigation simplicity and cognitive accessibility, enabling IDD users to engage with mobile applications effectively. However, their work was constrained by its focus on static applications, which lacked adaptability for varying learning levels and content types. Moreover, the proposed designs did not address dynamic educational needs, such as generating tailored quizzes or interactive games.

Our project builds on these insights by introducing a centralized hub for learning templates tailored to different difficulty levels and topics. This centralized system ensures flexibility, allowing educators to generate customizable applications that cater to the diverse needs of students with disabilities. For instance, features like voice-based navigation, customizable layouts, and integrated assistive tools enhance accessibility while maintaining simplicity. Additionally, our focus on dynamic content generation addresses the evolving needs of both students and educators, ensuring a personalized and inclusive learning experience.

Despite advancements in assistive technology, significant challenges persist. Adapting applications for users with varying degrees of cognitive and motor impairments requires extensive customization, and ensuring real-world effectiveness beyond experimental environments remains a critical hurdle.

Furthermore, the high costs and limited awareness surrounding assistive features have hindered widespread adoption. By incorporating universal design principles, such as speech-to-text capabilities and pictogram-based communication, our project seeks to overcome these barriers. Moreover, reducing dependency on caregivers for interface configuration ensures greater autonomy for users with disabilities.

In conclusion, while existing works like those of Hameed et al. and Borblik et al. provide valuable foundations, our project takes a more holistic approach. By integrating advanced software-based templates with inclusive design principles, it addresses the limitations of prior systems and demonstrates the potential of assistive technology to transform digital learning for users with disabilities.

2.4 LARGE LANGUAGE MODELS (LLMS) IN ASSISTIVE APPLICATIONS

Large Language Models (LLMs) have shown tremendous potential in generating customizable and context-specific content, making them valuable tools in accessibility and personalized learning domains. By leveraging their ability to process and generate natural language, LLMs can play a crucial role in creating educational content that meets diverse user needs. Recent advancements highlight the integration of LLMs with complementary technologies to enhance their applicability across various fields.

Zhang et al. (2023) [5] proposed an innovative approach that combines LLMs with knowledge graphs to generate simulation scenarios for military training. This integration enables the creation of detailed, context-aware content by using knowledge graphs to structure domain-specific data into entities and relationships. These structured datasets act as a foundation for LLMs to reason and generate actionable insights. The prompts, serving as a bridge between the two technologies, translate structured data into natural language outputs, facilitating

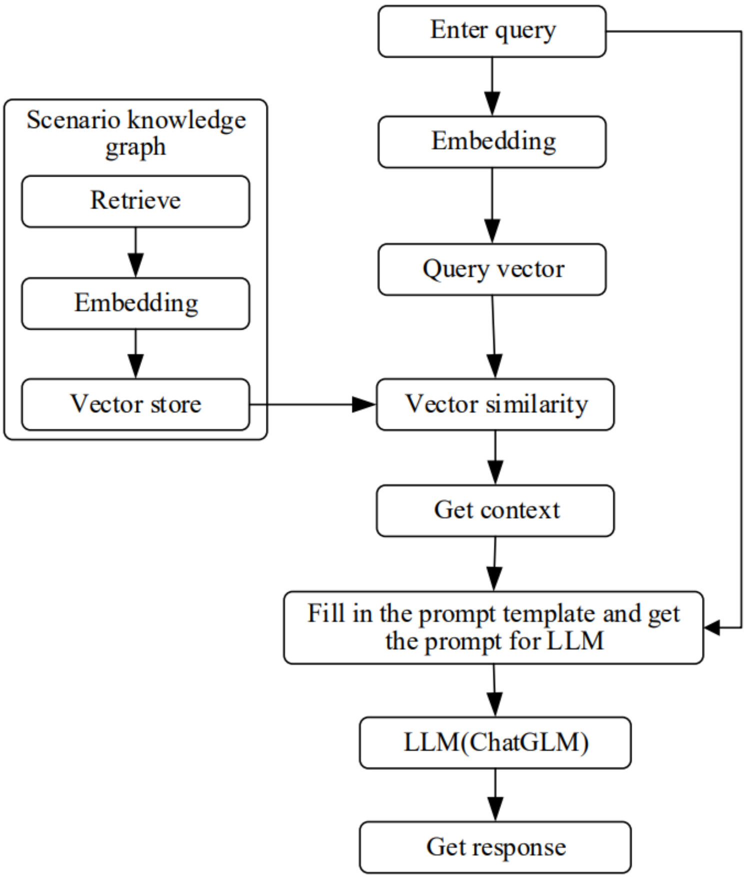

Figure 2.3: Integration of Knowledge Graph and LLM for Simulation Scenario Generation

seamless communication between systems. Figure 2.3 illustrates this integration, showcasing how knowledge graphs contribute precision and reliability while LLMs handle scalability and contextual reasoning. This methodology has been particularly effective in producing adaptive scenarios tailored to military and industrial training systems.

While Zhang et al.'s work underscores the strengths of combining LLMs with knowledge graphs, it also highlights certain challenges that limit its application in educational or assistive domains. The complexity of fusing the structured precision of knowledge graphs with the contextual adaptability of LLMs often results in integration bottlenecks. Additionally, ensuring accurate outputs requires extensive preprocessing and dataset alignment, especially in high-stakes fields like military training. Scalability remains a concern as input data volumes grow, demanding efficient handling without sacrificing speed or accuracy. Designing intuitive prompts to encapsulate complex domain knowledge further adds to the implementation challenges, necessitating constant refinement of both models and data inputs.

Our project adapts the principles of LLM integration to address gaps in accessibility and education, focusing on creating tailored templates for students with disabilities. Unlike Zhang et al.'s military-centric application, our approach leverages LLMs to generate content that aligns with diverse learning needs, incorporating customizable features such as dynamic difficulty adjustments and voice-based navigation. These educational templates are not only adaptive but also designed to be inclusive, empowering educators and students to interact with content intuitively. By prioritizing accessibility, our system demonstrates the versatility of LLMs in domains beyond simulation scenarios, bridging the gap between advanced language models and practical, user-centered applications.

Zhang et al.'s approach highlights the potential of LLMs for content generation but is not designed for educational applications or accessibility. Our project uses LLMs to generate language learning content, adapting the technology to enhance user experience for students with disabilities by enabling customized and accessible educational templates.

- The complexity involved in fusing knowledge graphs with LLMs stems from their inherently different data paradigms. Knowledge graphs emphasize precision and structure, whereas LLMs excel in contextual and probabilistic reasoning, leading to potential integration bottlenecks.
- Ensuring the accuracy and reliability of domain-specific outputs requires extensive preprocessing and alignment of datasets. The challenge is heightened in fields like military simulation, where incorrect inferences can result in critical errors.

- Scalability poses a significant hurdle, as the volume and variety of input data increase. Efficient handling of large datasets without compromising processing speed or accuracy remains an ongoing concern.
- Designing intuitive prompts for bridging knowledge graphs and LLMs is intricate. The effectiveness of this approach relies heavily on the ability to encapsulate complex domain knowledge within comprehensible and functional prompt structures.
- Addressing the limitations of LLMs, such as logical inconsistencies and insufficient domain knowledge, necessitates constant refinement of both the model and its training datasets.
- The process of adapting existing simulation frameworks to integrate LLMs and knowledge graphs often requires substantial re-engineering, which may be resource-intensive and time-consuming.

2.5 DESIGNING ENGAGING GAMES FOR EDUCATION: MOTIVATORS AND PRINCIPLES

Laine and Lindberg (2020) [6] conducted a systematic literature review to investigate the use of game motivators and design principles in educational contexts. Their study highlights the importance of learner engagement as a critical factor in educational success. Engagement, defined as the level of involvement in the learning process, can stem from extrinsic motivators (such as rewards and grades) or intrinsic motivators (like curiosity and challenge). The challenge lies in maintaining long-term engagement, especially when the learning content is inherently unappealing or demanding.

The study identifies two major contributions to educational game design: a taxonomy of 56 game motivators grouped into 14 classes, and a taxonomy of 54 educational game design principles grouped into 13 classes.

These taxonomies serve as practical toolkits for educators and game designers, helping them create motivating and effective educational games. Furthermore, the study emphasizes the difference between extrinsically and intrinsically motivated engagement, advocating for the use of game design elements that foster intrinsic motivation, as it has been shown to sustain long-term commitment and deeper learning outcomes.

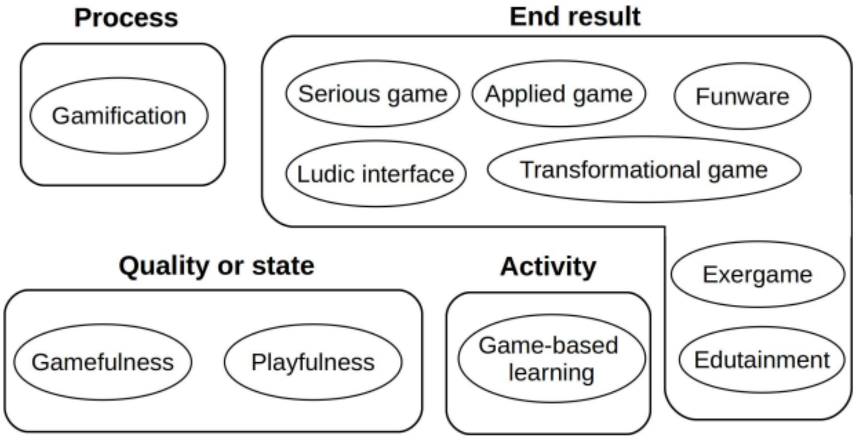

Figure 2.4: Classification of terms related to gamification.

Our project builds upon the principles highlighted by Laine and Lindberg by incorporating adaptive game motivators that align with the learners' engagement patterns. By leveraging interactive dialogue and gamified content, we aim to enhance student motivation through both extrinsic and intrinsic motivators, fostering long-term commitment to learning. Our framework emphasizes the development of interactive and immersive educational experiences, addressing the challenge of maintaining learner interest while delivering meaningful content.

2.6 INNOVATIVE GAME-BASED EDUCATIONAL APPLICATION FOR LEARNING

The integration of educational content with entertainment, known as ”edutainment,” has emerged as an effective strategy to engage learners. A 2021 study Poojary Shubham [7] presented at the International Conference on Computing, Communication and Networking Technologies (ICCCNT) introduced an innovative game-based educational application. By combining competitive elements with learning objectives, the application aimed to increase user motivation and retention—key factors for success in digital education.

However, implementing multiplayer functionality introduces unique challenges. Designing a game that balances engaging mechanics with educational content requires careful calibration, as prioritizing one over the other may diminish its effectiveness. Moreover, ensuring seamless synchronization and managing data privacy in multiplayer modes, especially for younger audiences, adds technical complexity. The paper emphasizes the importance of creating an intuitive user interface that is easy for students to navigate, ensuring the platform remains accessible to learners with varying levels of experience with digital games.

Our project takes inspiration from this approach by integrating interactive and engaging elements into accessible educational templates. While we do not focus on game-based learning, our customizable templates offer an engaging user experience tailored to diverse needs and abilities. For instance, the inclusion of voice prompts and simplified navigation makes the templates suitable for students with disabilities. Unlike game-centric platforms, which may not cater to all learners, our approach prioritizes inclusivity and adaptability, ensuring that users can engage in enjoyable, personalized learning experiences. By combining interactivity with accessibility, our project seeks to redefine educational tools for learners with diverse needs.

2.7 SUMMARY OF THE EXISTING WORK

The existing body of research highlights significant advancements in template-based code generation, no-code development platforms, and assistive technology for individuals with disabilities. Template-based code generators, such as the one developed by Uyanik and Şahin (2020) [1], demonstrated remarkable efficiency in automating repetitive coding tasks, achieving nearly a 98.95% reduction in development time. However, their application was primarily limited to enterprise systems, with restricted flexibility for customization and accessibility. Similarly, no-code development platforms like AUTOGEN STUDIO have enabled rapid prototyping and workflow creation but remain largely targeted toward technical users, with limited consideration for non-technical educators.

In the realm of assistive technology, research has focused on hardware-driven solutions, such as laser-based control systems and sensor-assisted input devices, to address physical disabilities. While these innovations provide valuable accessibility solutions, their reliance on specialized hardware limits scalability and adaptability. Furthermore, game-based educational applications have demonstrated the potential to enhance learning through interactive and engaging experiences. However, the integration of accessibility features, tailored educational content, and user-friendly interfaces remains a significant gap in existing solutions. Recent advancements in Large Language Models (LLMs) like GPT and LLaMA have shown promise in generating dynamic and context-specific content, but their application in inclusive learning environments is still in its infancy.

The analysis of these works highlights the need for a platform that bridges these gaps by integrating accessibility, user-friendly design, and dynamic content generation. This project builds upon the strengths of these studies while addressing their limitations, aiming to deliver an inclusive, customizable, and adaptive educational tool for diverse learning needs.

2.8 RESEARCH OBJECTIVES

The overarching goal of this project is to develop a comprehensive platform that empowers educators to create accessible and customizable educational applications, addressing the diverse needs of students with disabilities. To achieve this, the following research objectives have been defined:

- **Develop an Accessible Application Generation Platform:** Create a no-code environment that enables educators to design applications without requiring programming knowledge, thereby lowering the barrier to entry for non-technical users.

- **Leverage Large Language Models for Dynamic Content Generation:** Integrate LLMs like LLaMA with LangChain to generate context-specific and customizable templates, ensuring that educational content is tailored to individual learning levels and topics.

- **Incorporate Advanced Accessibility Features:** Design features such as single-click navigation, on-screen keyboards with timed traversal, voice-based input options, and sweep stick functionality to ensure usability for individuals with limited mobility or cognitive impairments.

- **Provide Versatile Learning Templates:** Offer a range of templates, including quiz-based modules, flashcard games, and interactive math exercises, catering to various educational objectives and learning styles.

- **Ensure Scalability and Adaptability:** Develop a platform architecture that supports scalability for diverse use cases, including real-time learning, adaptive difficulty settings, and integration with assistive technologies.

- **Validate Usability Through Extensive Testing:** Conduct rigorous testing to evaluate the platform's performance, user satisfaction, and accessibility under real-world conditions, ensuring that the solution meets the needs of both educators and students.

- **Promote Inclusivity in Digital Learning:** Align the platform with modern accessibility standards, such as WCAG and ADA, to create an inclusive learning environment that empowers students of all abilities to achieve their educational goals.

CHAPTER 3

SYSTEM DESIGN OF EDUCATION FRAMEWORK

3.1 OVERVIEW

The proposed project introduces a no-code educational framework designed for both students and educators. This platform leverages large language models (LLMs) to enable users to create customized learning applications without requiring programming expertise. Unlike traditional educational tools that often focus on static content delivery, this framework facilitates interactive, adaptive, and accessible learning experiences tailored to individual needs.

By integrating a user-friendly interface, the framework ensures seamless interaction for educators to generate educational applications dynamically. It supports essential features such as natural language-based content generation, template customization, and an accessibility hub to enhance usability. the framework incorporates AI-driven content suggestions, multilingual support, and real-time collaboration tools to enhance the learning experience. It ensures compatibility across multiple devices, allowing access through web and mobile platforms. The system empowers teachers to design and distribute interactive exercises, quizzes, and gamified learning experiences while providing students with an engaging and adaptive learning environment.

This chapter discusses the system's architecture, core modules, and data flow. It highlights the process of prompt-based application generation, template management, and accessibility integration. The subsequent sections outline the key functional components and their interactions, ensuring flexibility, scalability, and ease of use within educational contexts.

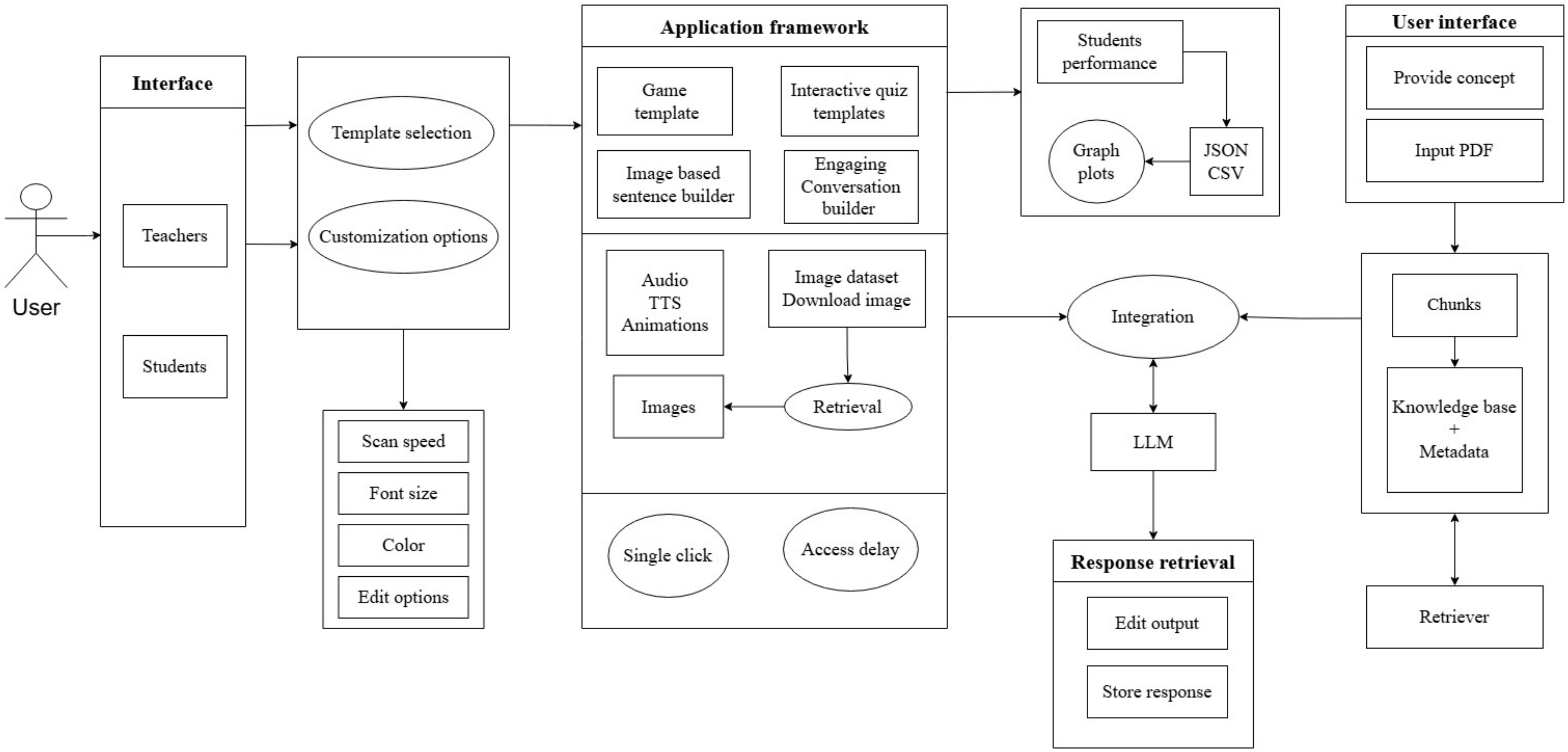

Figure 3.1: System Architecture of Application Generation

The architecture of the educational framework, as illustrated in Figure 3.1, is structured into interconnected modules that streamline content generation, accessibility enhancement, and deployment. The framework is designed to accommodate diverse educational needs, including interactive content, accessibility options, and real-time customization.

The process begins with the **Application Generator Interface**, where educators input natural language prompts or upload PDF documents. The system leverages **Training and Mapping Models** to process these inputs, utilizing large language models (LLMs) to generate structured and interactive educational content.

In **Phase 1**, the generated content is mapped to predefined templates through the **Template Management Module**, ensuring consistency in presentation and usability. Educators can customize these templates by selecting parameters such as difficulty level, accessibility options, and subject matter.

Phase 2 introduces an enhanced user interface for both teachers and students. It includes options for template selection, customization of color, font, and speed, as well as editing capabilities. The **Application Framework** supports diverse templates, such as:

- Game-based learning templates
- Interactive quiz templates
- Image-based sequence builders
- Engaging conversation builders

The framework also integrates advanced features such as audio text-to-speech (TTS), image dataset downloads, retrieval augmentation, single-click accessibility, and adjustable access delay.

The **Performance Analysis Module** visualizes student progress through graph plots and data exports in JSON and CSV formats. The **User Interface (Web UI)** supports input through natural language prompts, PDF uploads, and manual editing of generated outputs. The integrated knowledge base and metadata management facilitate efficient response storage and retrieval.

This holistic framework bridges accessibility and interactive learning, adapting seamlessly to various educational environments and user needs.

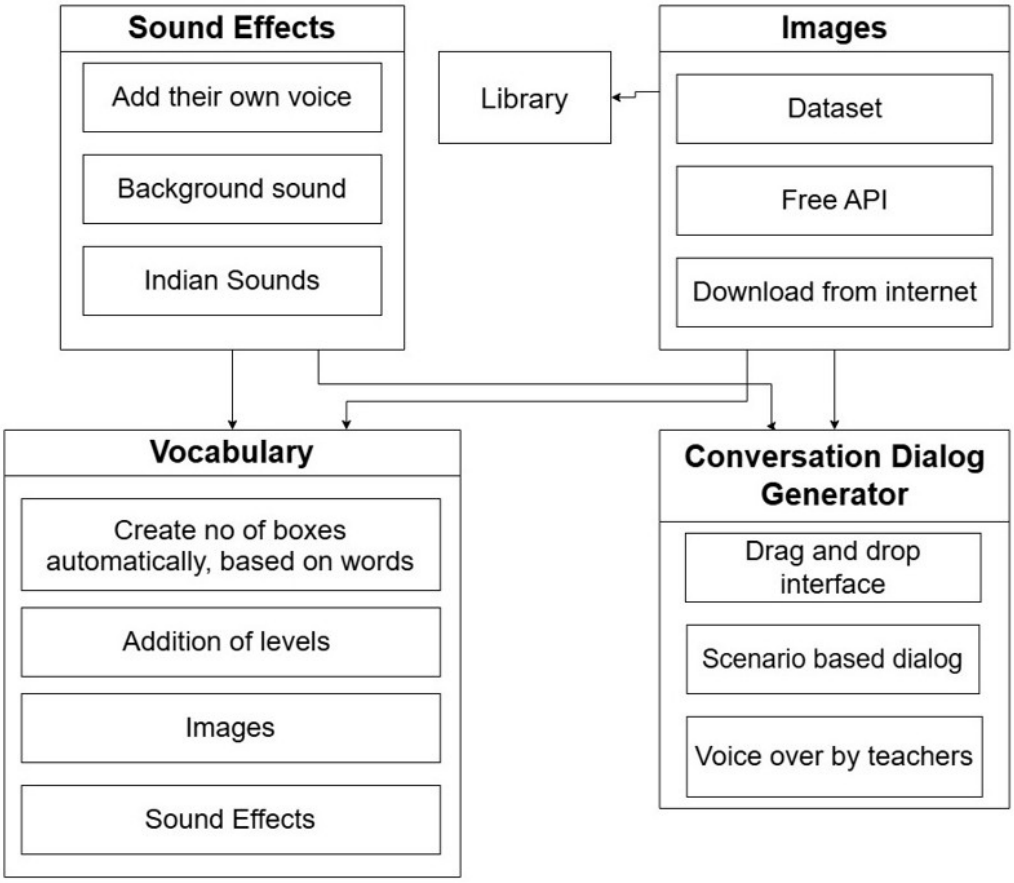

Figure 3.2: Framework Flow

3.1.1 Template Generation and Management

The system incorporates a Template Management Module, where templates serve as the foundation for content generation shown in the Table. Once prompts are processed by the LLM, they are mapped to predefined templates based on user requirements. These templates include elements such as quiz-based applications, single-click games, and customized learning exercises. Educators can select appropriate templates through the hub, and the system dynamically fills them with content, including text, images, and interactive components. The system also supports the addition of accessibility elements, such as text-to-speech features and hover-based interactions, ensuring compatibility with diverse educational needs. This modular approach simplifies the integration of generated content into usable applications, maintaining consistency and scalability.

3.2 QUIZ TEMPLATES - MODULE

This section introduces various quiz-based learning templates that allow educators to create interactive assessments tailored to student needs. The templates support customizable UI elements, dynamic content generation, and voice-assisted features to enhance accessibility. The following table (3.1) summarizes the various quiz templates and their functionalities.

Table 3.1: Quiz Templates and Their Functionalities

Template Type	**Features and Customization**
MCQ and Fillups Quiz	Dynamic color selection for question and answer boxes, customizable font style and size, voice prompt, automatic question traversal on correct answer
Interactive Dialogue Builder	Context-based conversation generation, dynamic character image retrieval, teacher-controlled font size and dialogue timing
Story-Based Learning	AI-generated stories based on prompts, embedded quizzes, and branching scenarios for decision-making

3.2.1 MCQ and Fillups Quiz

The MCQ and Fillups Quiz module has been significantly upgraded from the previous phase to enhance interactivity and customization. This time, we have introduced image support, allowing teachers to include images within questions using a simple drag-and-drop feature from the image selector. The platform also offers flexibility in dynamically adjusting the number of questions and options based on the teacher's requirements.

Once the quiz content is set up, clicking the 'Save' button stores the dynamically generated quiz content. The saved quiz is then transformed into an interactive UI, offering a responsive and engaging experience for students. Additionally, a customization panel allows teachers to adjust visual settings, including font color, background color, and scan speed, ensuring a user-friendly and accessible interface.

Table 3.2: Upgraded Features of MCQ and Fillups Quiz Template

Feature	**Description**
Dynamic Question and Option Count	Teachers can increase or decrease the number of questions and options as needed.
Image Integration	Supports drag-and-drop image selection to enhance question clarity.
Customization Panel	Allows changes in font color, background color, and scan speed.
Interactive UI	Transforms saved quiz content into an engaging, responsive interface.
Accessibility Features	Includes voice prompts and automatic question traversal upon correct answers.

3.2.2 Interactive Dialogue Builder

The **Interactive Dialogue Builder** module 3.3 is designed to generate **context-based conversations**, allowing students to engage in **real-world scenarios** such as **doctor-patient interactions** or **customer-service dialogues**. Leveraging refined **LLMs**, it produces interactive **Indian-accented dialogues** that mimic natural communication styles. The language models are fine-tuned with carefully crafted prompts to ensure realistic and context-appropriate responses. One of the unique features of this module is **dynamic character image retrieval**, where **character images** are automatically generated based on the prompt, offering a more immersive experience. These images are **freely available** and seamlessly integrated into the conversation flow. Additionally, the module allows educators to specify the **context** of the interaction, such as giving directions or discussing a problem. To further enhance engagement and accessibility, teachers have complete control over **customization options**, including **font size**, **dialogue timing**, and advanced **voice settings**. The module supports over **15 different voice profiles** with customizable **pitch and speed**, allowing educators to tailor the audio experience to suit different learning needs. Additionally, the module features a **history view** that enables users to revisit previous conversations, making

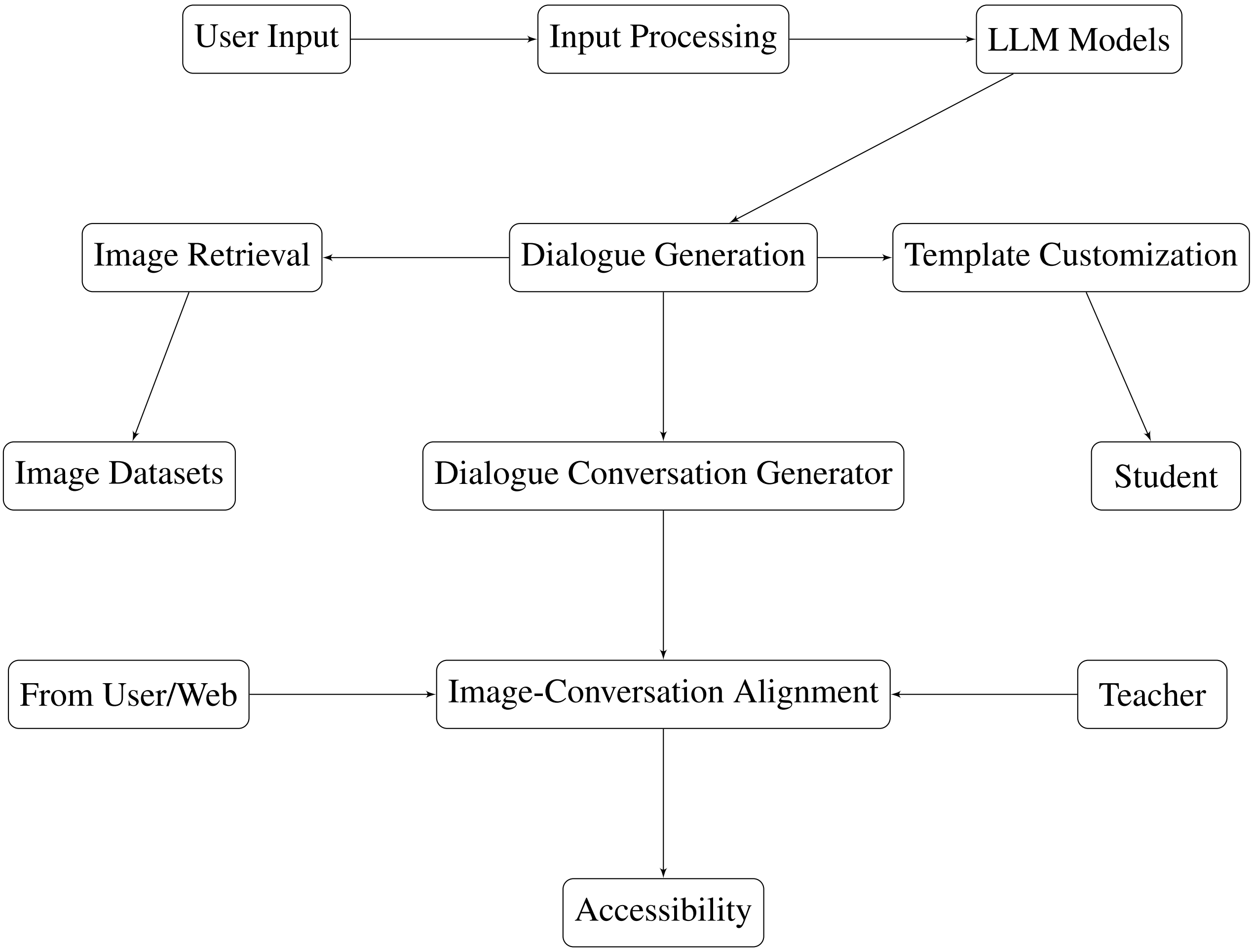

Figure 3.3: Flowchart of the Educational Framework Architecture

it useful for revision and practice. Educators can also **prompt specific characters or situations**, making the dialogue generation flexible and targeted. These comprehensive features make the **Interactive Dialogue Builder** an engaging tool for developing practical communication skills.

3.2.3 Story-Based Learning

AI-generated stories are created based on given prompts, integrating embedded quizzes and branching scenarios where students make choices that influence the storyline. This gamified storytelling approach fosters engagement and critical thinking skills.

3.3 AI INTEGRATION FOR LEARNING

AI plays a significant role in personalizing education by dynamically generating content, automating assessments, and enhancing accessibility. This section describes AI-powered modules that support different learning styles.

Table 3.3: AI-Based Learning Modules

Module	**AI-Driven Features**
Code Generation Assistant	Supports Python (Tkinter-based apps), real-time preview, export as executable files
Picture-Based Learning	Auto-generated images with labels, interactive touch-based selection, multilingual text-to-speech support
AI-Powered Content Generator	Generates interactive learning materials, quiz questions, and explanations based on topics

3.3.1 Code Generation Assistant

This AI-driven tool helps students and educators create simple applications without coding expertise. It supports Python-based Tkinter applications, provides real-time previews, and allows exporting applications as executable files.

3.3.2 Picture-Based Learning

AI automatically generates images with labels to support visual learning. Students interact with touch-based selection methods, and multilingual text-to-speech support ensures accessibility for diverse learners.

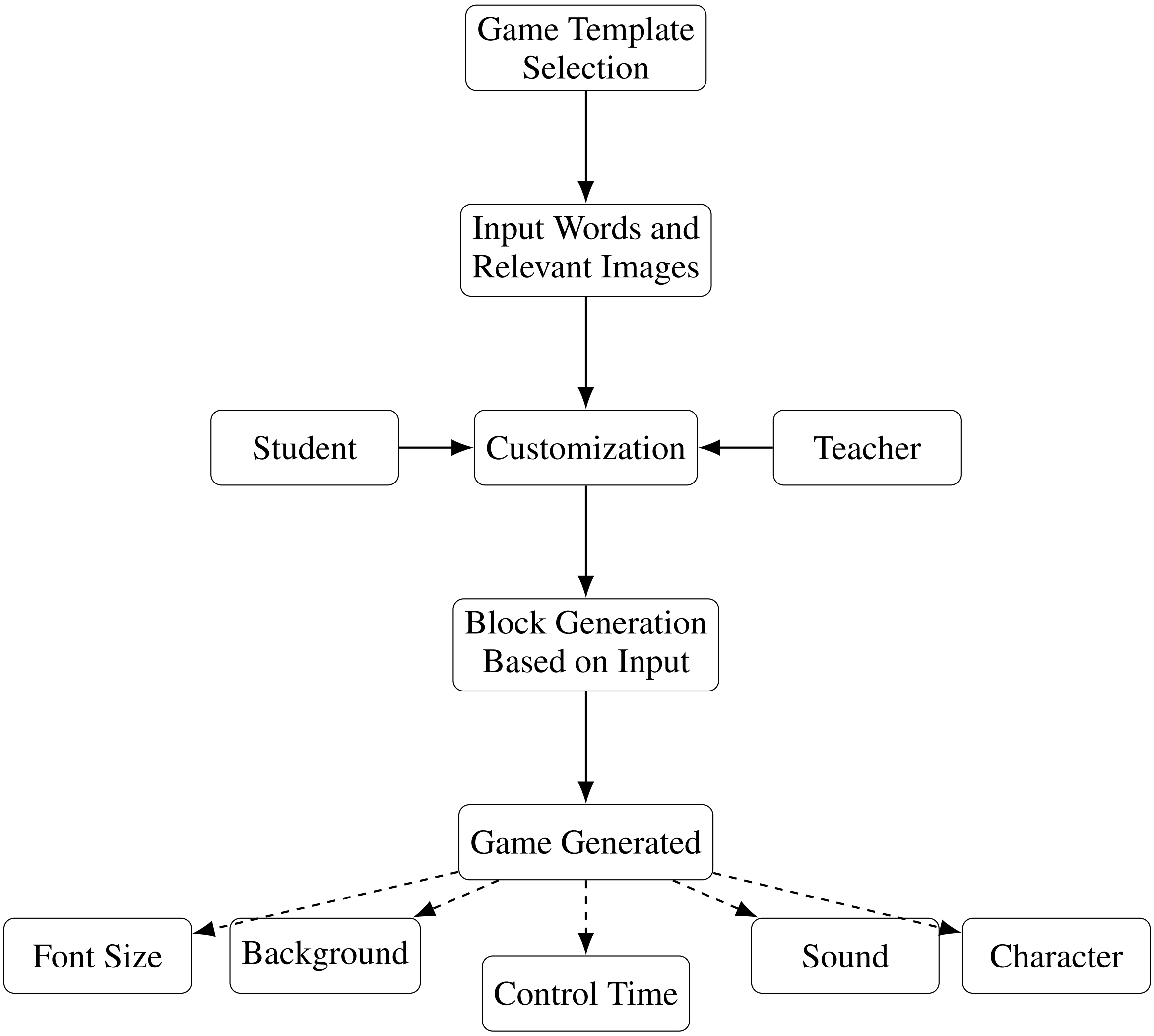

Figure 3.4: Image based learning

3.3.3 AI-Powered Content Generator

This module generates educational materials such as interactive quizzes, explanations, and structured lesson plans based on given topics. It helps educators save time while ensuring high-quality content generation.

3.4 EDUCATIONAL GAME FEATURES

Gamification enhances learning by making educational activities more engaging and interactive. This section explores various game-based templates designed to reinforce concepts through play.

3.4.1 Math Game (Spaceship)

Students solve arithmetic problems to control a spaceship and progress through levels. The game includes options for addition, subtraction, multiplication, and division, as well as adjustable difficulty levels, timers, and spaceship customization.

3.4.2 Flashcard Learning Game

This game-based learning approach allows students to memorize topics such as animals and birds through interactive flashcards. Pop-ups provide feedback on missed items, and integrated audio or voice support reinforces learning through repetition.

3.4.3 Puzzle-Based Learning

AI-generated puzzles provide an engaging problem-solving experience. Drag-and-drop interactions, adaptive difficulty levels, and hint mechanisms help students build critical thinking and logical reasoning skills.

Table 3.4: Gamified Learning Templates

Game Type	Features and Customization
Math Game (Spaceship)	Options for addition, subtraction, multiplication, division; level upgrades, timer/delay settings, spaceship vehicle selection via API/model
Flashcard Learning Game	Topics like animals, birds; pop-ups for missed items, audio/voice options for learning reinforcement
Puzzle-Based Learning	AI-generated puzzles, drag-and-drop interactions, and hints for problem-solving

3.5 APPLICATION GENERATOR INTERFACE

The Application Generator Interface shown in the Figure 3.5 serves as the central hub for educators to create and customize applications tailored to the needs of students with disabilities. Using this interface, educators can input natural language prompts, which the system then processes to generate educational templates. Constraints are handled directly within the prompt-processing module to ensure compatibility with accessibility requirements. Customizable features such as font size, color selection, and UI layout allow educators to design applications that meet specific accessibility needs.

3.6 PROMPT PROCESSING AND TRAINING

In the prompt processing module depicted in Figure 3.6, user inputs are mapped and preprocessed for compatibility with large language models (LLMs). This phase extracts essential keywords, context, and accessibility preferences. The model interprets phrases such as ”enable automatic navigation with voice” or ”single-click navigation” and maps them to corresponding accessibility features in the template. A custom-trained LLM further enhances this process, enabling

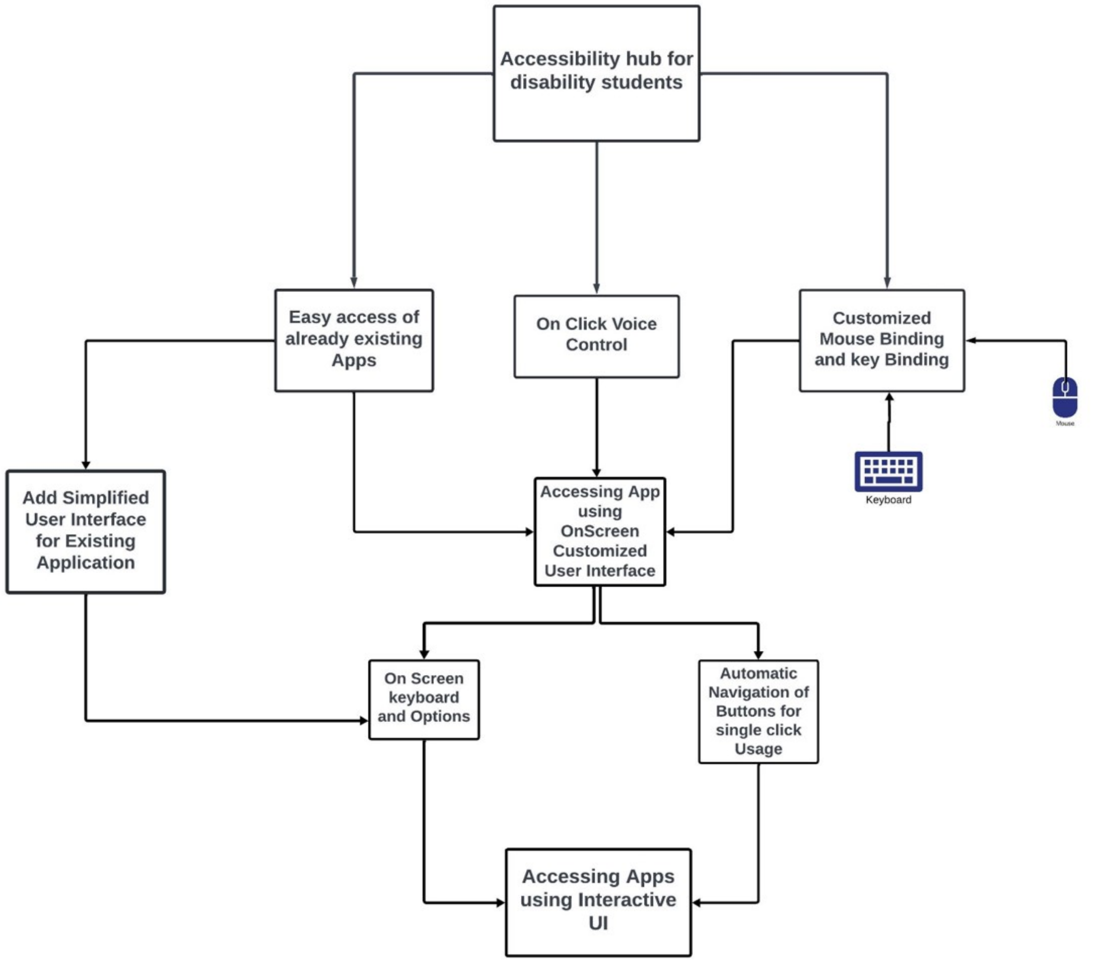
Accessibility hub for disability students
Easy access of already existing Apps
On Click Voice Control
Customized Mouse Binding and key Binding
Mouse
Keyboard
Add Simplified User Interface for Existing Application
Accessing App using OnScreen Customized User Interface
On Screen keyboard and Options
Automatic Navigation of Buttons for single click Usage
Accessing Apps using Interactive UI

Figure 3.5: Diagram of the Application Generator Interface showcasing customizable options.

the system to interpret nuanced instructional needs and automatically align them with template options.

3.7 LARGE LANGUAGE MODELS (LLMS) FOR CONTENT AND CODE GENERATION

In modern systems utilizing Large Language Models (LLMs), data flow plays a critical role in ensuring efficient query handling and content generation. As illustrated in Figure 3.7, data originates from multiple sources, including structured data such as databases, unstructured data like documents, and programmatically fetched data via APIs. These sources are consolidated into an *Index*, which acts as an intermediary for organizing and optimizing data retrieval. The system enables users to send queries, which are processed by retrieving

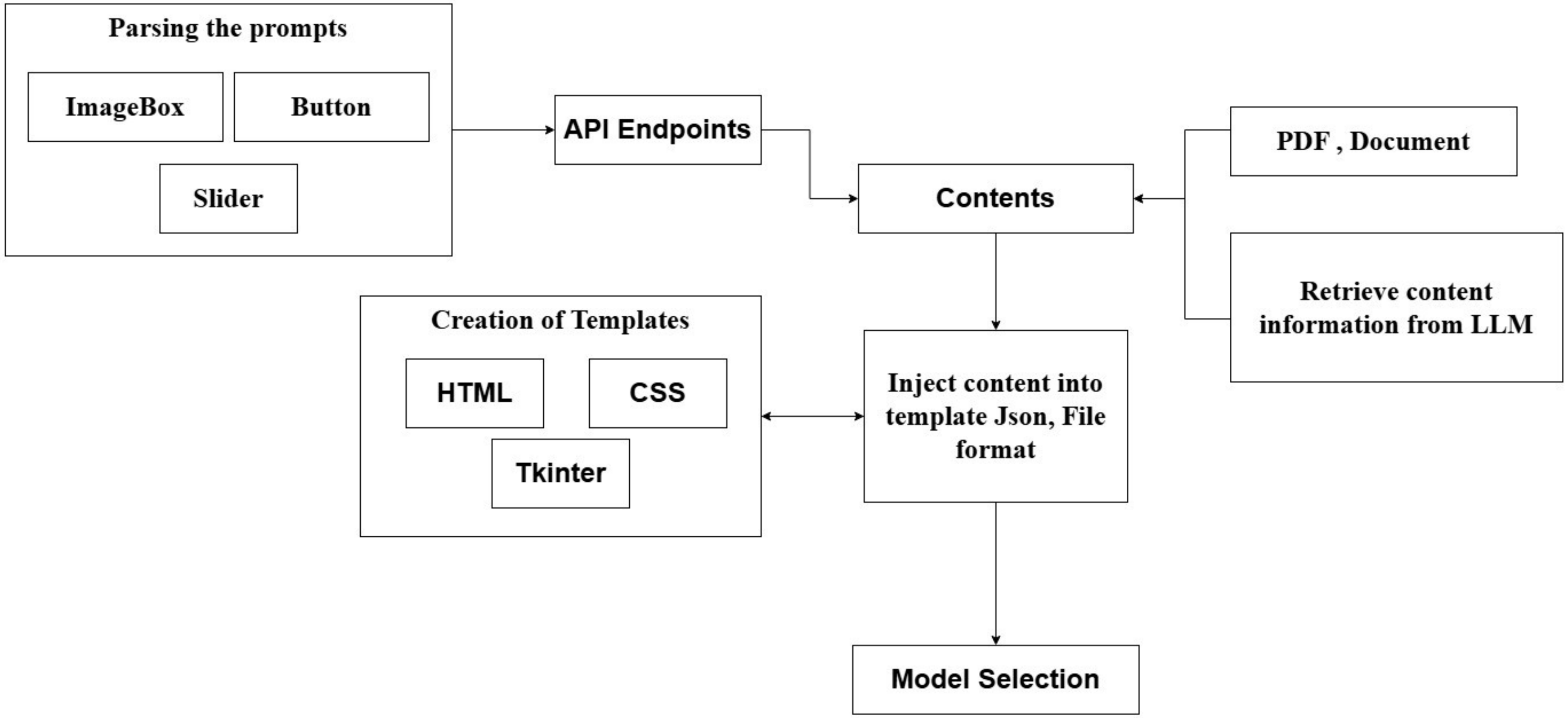

Figure 3.6: Flowchart of Prompt Processing and LLM Training for Template Generation

relevant information from the index. This process ensures that only the necessary context is combined with the query to form an optimized *prompt* for the LLM.

The LLaMA model is employed to generate educational content based on specific topics and difficulty levels provided by the educator. For instance, when given a topic like ”birds” along with a specified difficulty level, the LLM produces a set of questions and answers that align with these parameters. This content is then formatted into JSON for easy integration with the platform's templates. While customization options such as dynamic color selection must be manually set by the user, the LLM facilitates content creation, making it easier for educators to populate applications with relevant and structured information tailored to their instructional needs.

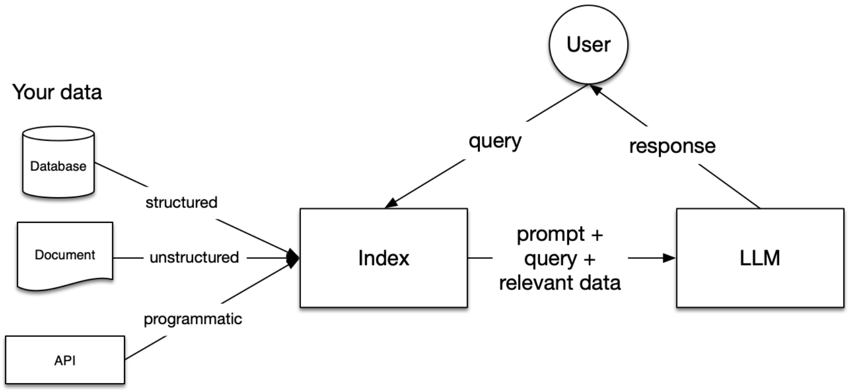

Figure 3.7: Data Retrieval and LLM-Based Content Generation Workflow

3.8 TEMPLATE SELECTION AND CUSTOMIZATION

Educators can choose from a variety of pre-designed templates tailored to various instructional goals shown in the Figure 3.8. Each template includes accessibility options like text-to-speech features, color settings, and font adjustments. Templates are organized by difficulty level (easy, medium, hard) to support differentiated learning. This system ensures flexibility, allowing educators to select options for question types, UI layout, and accessibility preferences.

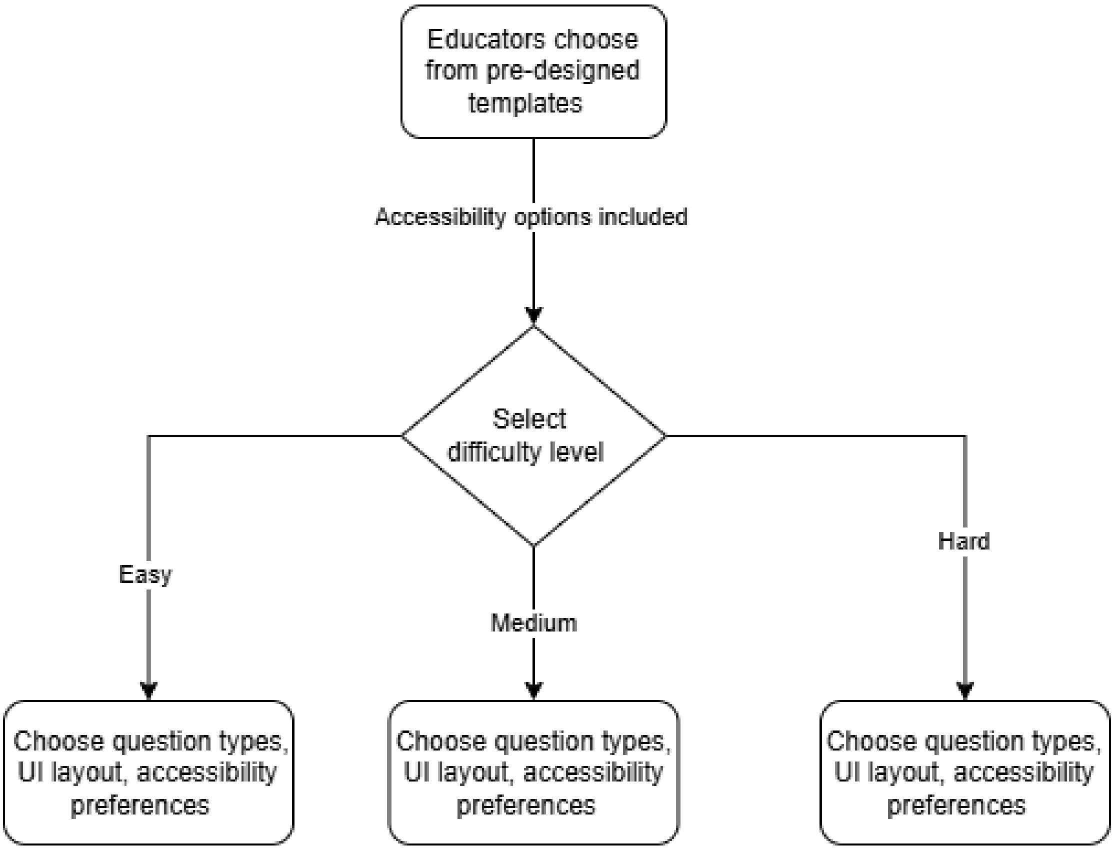
Educators choose from pre-designed templates
Accessibility options included
Select difficulty level
Easy
Medium
Hard
Choose question types, UI layout, accessibility preferences
Choose question types, UI layout, accessibility preferences
Choose question types, UI layout, accessibility preferences

Figure 3.8: Template Selection Interface with Accessibility Features.

3.9 DEBUGGING AND TESTING OF GENERATED APPLICATIONS

The system in the Figure **??** includes an integrated testing environment where educators can preview applications, adjust settings, and ensure that all features function correctly. text-to-speech features, hover-triggered navigation,

dynamic color changes are validated within this testing environment. Additionally, users can simulate various accessibility settings to ensure compatibility with student requirements, enhancing the reliability of generated applications.

Once the application content is generated, the teacher can preview the application and has the liberty to correct the errors and modify the styling options according to the requirements. The teacher downloads the Llama model locally on their machine, previews the application, and then finalizes the complete setup. This web application is then integrated into the hub, allowing the teacher to update and add new tests or gamified learning content to the hub whenever needed.

3.10 UI CUSTOMIZATION FOR ACCESSIBILITY

Table 3.5 illustrates, A key feature of this system is the adaptability of the user interface (UI) to accommodate individual user preferences. Customizable options include font resizing, color adjustments, and voice-assistance toggles. Each of these features aims to improve the accessibility of applications for students with various impairments. The UI has been designed to work with single-click and provide customized styling, providing an optimal experience.

Table 3.5: Accessibility Features Across Templates

Feature	**Description**	**Templates Supported**
Dynamic Color Selection	Color customization for elements like question boxes and navigation buttons	MCQ Quiz, Math Game
Custom Font Size	Adjusts text size to enhance readability	All templates
Single-Click Navigation	Simplified navigation for easier interaction	Flashcard Learning Game, Story Builder

3.11 SOFTWARE REQUIREMENTS

The success of this project heavily depends on a well-defined and efficient technology stack that can support both the backend functionalities and the frontend user experience. To build a system that is both accessible and scalable, a combination of powerful programming languages, libraries, and frameworks are chosen. These tools work synergistically to create an application that provides seamless integration between content generation, accessibility features, and user interaction.

In this section, we will outline the programming languages, libraries, and development tools that form the foundation of the platform, ensuring its smooth operation and responsiveness across different user needs. The choice of technologies is made with an emphasis on performance, ease of development, and accessibility, which are key factors in ensuring that the platform serves educators and students effectively.

3.11.1 Programming Language

The programming languages chosen for this project are Python and JavaScript. Python is used primarily for the backend logic and accessibility tools, whereas JavaScript is used extensively on the frontend to ensure a dynamic, responsive, and interactive user interface.

On the frontend, JavaScript, along with React JS and Vite.js, forms the core technology stack. Vite.js, a fast development build tool, is employed to enhance the React development experience by providing faster hot module replacement and optimized build performance. React JS is used for building the frontend components and managing the dynamic UI elements.

Python is employed for the backend, providing the foundation for the application’s core functionalities. Python’s simplicity and the vast ecosystem of libraries make it an ideal choice for backend services, particularly in integrating machine learning models, managing the accessibility features, and controlling system-level operations.

3.11.2 Required Libraries

The system makes extensive use of several Python libraries for both backend logic and integration with accessibility features. These libraries help in managing data processing, generating content, controlling accessibility features, and enhancing the user experience. tabularx

Table 3.6: Frontend Technologies Used

Technology	Description
React JS	React is the core library used to build the interactive and dynamic components of the frontend. It enables efficient rendering of UI elements and allows for a modular approach to creating quiz templates, interactive games, and customizable features for teachers and students.
Tailwind CSS	Tailwind CSS is used for styling the frontend components. It provides utility classes that allow for rapid design changes and customization without writing custom CSS. It enables flexible, responsive layouts and easy customization for visual elements like quiz themes, font sizes, and colors.
Vite	Vite is a build tool that supports modern JavaScript features and offers hot module replacement (HMR). It speeds up the build process and allows for instant feedback during development, improving productivity.
Speech Synthesis	The Web Speech API, particularly the Speech Synthesis feature, is used to convert text into speech, making the application more accessible for AAC users. It reads aloud quiz questions, options, and answers, with customization options for teachers.

Below are some of the key libraries used for backend logic and integration with the platform's accessibility features:

3.11.3 Development Tools

Several tools are utilized to streamline the development process for both frontend and backend:

Table 3.7: Backend Technologies Used

Technology	Description
Flask	Flask is a lightweight Python web framework used for building the backend services and APIs. Flask handles HTTP requests and facilitates communication between the frontend and backend.
Python	Python is used for backend development and tasks such as data processing, integration with AI models, and managing quiz content.
Llama Model	Llama is a large language model (LLM) used to generate answers, hints, and educational content, integrated into the Retrieval-Augmented Generation (RAG) pipeline to assist in dynamic content generation.
JSON	JSON (JavaScript Object Notation) is used to store and exchange data between the frontend and backend, ensuring smooth data handling and compatibility.
Chroma DB	Chroma DB is a vector database used for semantic search and RAG applications, enhancing retrieval capabilities for accurate responses, especially when integrating AI-based features.
Langchain	Langchain is a framework designed for building applications using language models, aiding in the creation of RAG pipelines and enabling seamless integration between the Llama model and Chroma DB for accurate quiz content generation.

- **Vite.js:** A build tool that serves as the backbone of the frontend development environment. It provides a fast and optimized workflow for React projects, ensuring that the developer experience is smooth and efficient.

- **Node.js and npm:** Node.js is the runtime used to execute JavaScript code outside the browser, and npm (Node Package Manager) is used to manage the JavaScript dependencies, including React, Vite, and other frontend libraries.

- **Visual Studio Code (VS Code):** A powerful code editor used for writing and debugging both Python and JavaScript code. It is equipped with extensions for React, Python, and debugging tools, making it a versatile environment for full-stack development.
- **Flask Development Server:** Flask is used to develop RESTful APIs and to serve the backend logic. The Flask development server runs the backend and communicates with the frontend via HTTP requests.

The development tools listed above help provide an efficient and seamless workflow for building and testing the platform. These tools contribute to a more streamlined process, enhancing both frontend and backend development while maintaining the project's performance and scalability.

REFERENCES

[1] B. Uyanik and V. H. Şahin. A template-based code generator for web applications. *Turkish Journal of Electrical Engineering and Computer Science*, May 2020.

[2] Chen J. Bansal G. Syed S. Fourney A. Zhu E. Wang C. Dibia, V. and S. Amershi. Autogen studio: A no-code developer tool for building and debugging multi-agent systems. *arXiv preprint*, arXiv:2408.15247, August 2024.

[3] Ali S. M. Hameed, K. and U. Ali. An approach to design keyboard and mouse assisting device for handicap users. In *IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pages 1–6, Toronto, ON, Canada, 2022.

[4] Shabalina O. Kultsova M. Pidoprigora A. Borblik, J. and R. Romanenko. Assistive technology software for people with intellectual or development disabilities: Design of user interfaces for mobile applications. In *6th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–6, 2015.

[5] Zhang Y. Chu M. Yang S. Zhang, J. and T. Zu. A llm-based simulation scenario aided generation method. In *IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 1350–1354, 2023.

[6] Teemu H. Laine and Renny S. N. Lindberg. Designing engaging games for education: A systematic literature review on game motivators and design principles. *IEEE Transactions on Learning Technologies*, 13(4):804–821, 2020.

[7] Poojary Shubham Savitha A Shenov Yahya M H I Asadi, C Gagan Babu. Innovative game based educational application for learning. In *12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6, Kharagpur, India, 2021.

[8] A. Kumar and S. Roy. Innovative game-based educational application for learning. In *12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6, Kharagpur, India, 2021.

[9] Smith A. Johnson, M. and J. Clark. Ai-based learning platforms: A case study in education technology. *International Journal of Educational Technology*, 12(4):215–223, April 2024.

[10] R. Thompson. *Advances in Educational Assistive Technologies*. Springer, Berlin, Germany, 2023.

[11] K. Williams and L. Brown. Designing interactive learning systems for children with disabilities. *Journal of Interactive Learning Environments*, 31(2):132–148, July 2023.

[12] H. Liu and T. Zhang. Creating accessible educational content for all students: A design framework. In *Proceedings of the 9th International Conference on Accessibility and Usability*, pages 200–207, New York, NY, USA, 2022.

[13] Patel R. Smith, G. and S. Kumar. An interactive game-based learning model for special education. In *Proceedings of the 8th International Conference on Special Education Technology*, pages 45–52, Boston, MA, USA, 2021.

[14] Harris B. Miller, D. and P. Turner. Leveraging ai in personalized education for students with disabilities. *Journal of Artificial Intelligence in Education*, 28(3):323–341, September 2022.

[15] M. Brown and P. Williams. *Assistive Technology in the Classroom: Enhancing Learning for Students with Special Needs*. Routledge, London, UK, 2020.

[16] J. Lee and S. Park. Enhancing accessibility in digital learning platforms: A template-based approach. *Journal of Educational Technology*, 43(2):145–156, April 2022.

[17] A. R. Smith and M. Johnson. Automating interface generation for accessible education. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 88–97, Minneapolis, MN, USA, 2020.

[18] T. Almeida and R. Costa. Designing educational apps for cognitive accessibility. *Journal of Inclusive Education Research*, 29:214–222, September 2019.

[19] L. Garcia and E. Chang. Integrating accessibility features in web applications using code templates. In *Proceedings of the Web for All Conference (W4A)*, pages 35–42, Ljubljana, Slovenia, 2021.

[20] T. H. Pham and L. D. Nguyen. Template-based tools for scalable educational content creation. *International Journal of Digital Education*, 18(3):301–310, June 2023.

[21] Lee W. Schmidt, H. and K. Brown. Developing accessible educational software using modular templates. In *2022 IEEE Conference on Human-Computer Interaction (CHI)*, pages 123–130, New Orleans, LA, USA, 2022.

[22] Lee H. Martinez, D. and A. Williams. Template-driven development for gamified learning applications. In *IEEE International Conference on Educational Technologies (EduTech)*, pages 45–52, Berlin, Germany, 2021.

[23] P. Rogers and L. Thomson. A framework for template-based adaptive learning systems. *Journal of Computer-Assisted Learning*, 39(1):77–89, January 2023.

[24] M. Zhang and F. Yu. Challenges and opportunities in template-based learning app development. In *Proceedings of the 2020 IEEE International Conference on Educational Technology*, pages 67–74, Seoul, South Korea, 2020.

[25] C. Lopez and G. Fernandez. Adaptive learning with code templates: A new approach to educational technology. *Computers in Human Behavior*, 139:107–116, May 2023.

[26] Y. Kim and X. Li. Building personalized learning applications using template-based code generation. In *2021 International Conference on Learning Analytics and Knowledge (LAK)*, pages 29–35, New York, NY, USA, 2021.