International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 12 | Dec 2024 www.irjet.net p-ISSN: 2395-0072

Digital Guide: A Smart Study Planner

Prof. Ashwini T N
Dept. of CS & E
Bangalore Institute of Technology Bangalore, India

Harsh Kumar
Dept. of CS & E
Bangalore Institute of Technology Bangalore, India

Anshu Kumar Gupta
Dept. of CS & E
Bangalore Institute of Technology Bangalore, India

Lakshya Mahawar
Dept. of CS & E
Bangalore Institute of Technology Bangalore, India

Aditya Raj
Dept. of CS & E
Bangalore Institute of Technology Bangalore, India

Abstract— Digital Guide is a smart study planner that merges the gap in the academic journey by addressing key challenges faced by students, educators, and institutions.By creating a centralized hub, the solution minimizes confusion, enhances learning efficiency, and empowers users to make informed decisions. For students and educators it aids in summarizing content and discussions; and for institutions, it aligns students' interests with appropriate courses. The system's intuitive interface and ensure seamless navigation and tailored experiences.

This paper presents the system architecture, evaluation methodology, and empirical results, contributing to the growing field of adaptive educational technology.Also introducing the mental health chatbot that significantly improves learning outcomes while reducing mental load in decision-making.

Keywords— digital guide,learning efficiency, mental health educational technology,intelligent tutoring system, student engagement metrics.

I. INTRODUCTION

In the era of information overload, students often struggle to identify relevant study materials, choose appropriate academic domains, and efficiently revise large volumes of content. These challenges lead to inefficiencies, confusion, and missed opportunities in personalizing education. Existing systems fall short in addressing these issues comprehensively, lacking centralized platforms that integrate resource curation, domain exploration, and efficient content revision.

The proposed system, *Digital Guide: A Smart Study Planner*, aims to bridge this gap by providing a unified platform designed to cater to the diverse academic needs of students. It offers a comprehensive set of tools like text summarization, video content summarization for enhanced study efficiency, reliable study material curation to reduce search time, and domain exploration facilitation to guide specialization choices.

Additionally, it incorporates personalized study schedules, adapting to individual learning paces and preferences, ensuring maximum productivity.

By integrating video summarization, resource curation, and domain exploration tools within a single platform, Digital Guide represents a significant advancement in educational technology. The system not only addresses current challenges in academic resource management but also provides a scalable foundation for future developments in personalized learning and academic guidance. Our findings suggest that this approach can significantly enhance the educational experience for all stakeholders while promoting more efficient and effective learning practices.

This paper explores the development and implementation of the *Digital Guide*, emphasizing its potential to revolutionize the academic journey by reducing inefficiencies and empowering students, educators, and institutions with tailored tools and insights for effective education planning.

The objective of *Digital Guide: A Smart Study Planner*, is to reduce the time students spend searching for quality materials and to enhance revision efficiency through a video summarization and text summarization tool that highlights key concepts from extensive content and the specialized curated content with experience and analyzing the challenges faced by student helps to understand the concept better along with daily motivation and helping students on their mental health by integrating a personalized chatbot ,that help them to cope up with current situation through carefully analyzed responses.

II. RELATED WORK

A. Literature Review

Recent research explores various approaches to developing

e-learning platforms. [1]Picchetola and Dieckbuch-Helmfeld (2020) focused on interactive learning methods like the enactivism approach, which enhances student engagement but faces challenges in classroom settings.[2] Randhavane and Gaikwad (2024) showed how AI-based feedback systems improve personalized learning by adapting to cultural and linguistic differences

The integration of artificial intelligence and machine learning has emerged as a crucial development in e-learning platforms.[4] Xu and Yu (2024) demonstrate how resource scheduling technology can enhance learning experiences through automated resource allocation, though highlighting computational complexity as a significant challenge. This is complemented by [10] research on machine learning's impact on student behavior, which emphasizes personalization benefits while raising important concerns about data privacy and algorithmic bias.

Several studies address specific implementation challenges in different educational contexts. [6] provide valuable insights into e-learning adoption challenges through data analysis, specifically examining institutional barriers. The geographical limitations of implementation are further explored by [7] ,highlighting infrastructure challenges such as internet connectivity and feedback mechanisms in specific regional contexts.

Research increasingly focuses on improving platform usability. [8] analyzed how higher education faculty use e-learning tools, emphasizing the importance of user-friendly designs. Similarly, [9] suggested redesigning platforms to meet diverse user needs, ensuring a smooth experience for students and educators alike.

Lastly, studies by Ayachi (2023)[3] on platform-driven English language learning and Ruiz et al. (2024)[5] on mathematics teaching platforms demonstrate how specialized applications can achieve significant improvements in specific subject areas. These studies highlight both the potential for targeted solutions and the challenges of scaling such approaches across broader educational contexts.

The literature indicates progress in AI-powered e-learning and specialized tools while pointing out challenges like technical complexity, accessibility, and data security.

B. Existing Solutions and Limitations

Existing AI driven e-learning platforms still face challenges that affect their broad applicability. Their AI-driven personalized learning features rely on high-quality data, which limits their effectiveness for diverse cultural and linguistic audiences, sometimes resulting in recommendations that are not universally relevant or accessible. .

Another notable problem is with platforms focused on specific subjects, like Khan Academy, excel in providing in-depth resources for fields such as math and language. However, they lack cross-disciplinary support, forcing students studying multiple subjects to juggle between platforms, which disrupts the cohesion of their learning experience.

Some AI summarization tools save students time by highlighting key ideas. However, these tools often struggle with academic-specific content that requires nuanced understanding, sometimes leading to incomplete or overly simplified summaries, which can limit their educational effectiveness.

Existing platforms demonstrate a clear need for a more integrated, user-friendly, and adaptable e-learning environment that can serve multiple domains seamlessly. Addressing issues like scalability and user accessibility is essential for an e-learning platform to fully meet the evolving needs of students and educators alike. By bridging these gaps, future digital learning solutions can enhance the educational experience, making it not only more personalized but also widely accessible and secure.

III. SYSTEM DESIGN

A. Sequence diagram

The sequence for exploring a relevant course is as follows:

1. The user initiates a registration for a portal int the website.
2. The website gives the option to select the domain of interest and his level of knowledge.
3. This data is stored in database and backend server shows the relevant course to the user from the predefined dataset.
4. The user can start learning the module also can track his /her progress and streaks.
5. The user can also make notes of it and with each course content articles is also provided

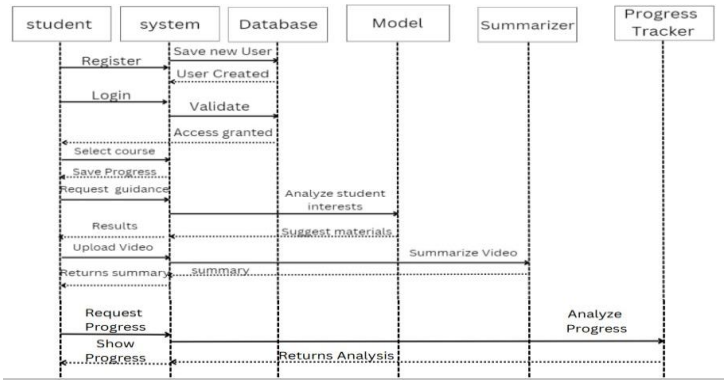

Fig. 1. Sequence diagram for exploring relevant course.

The sequence for tracking progress and revising the video is as follows:

1. The user can query for tracking his progress on ongoing module.

A. Feature Technology

The proposed system integrates the following technologies to achieve a robust and decentralized event promotion platform:

1. *Express. js:* A Node.js framework that simplifies the process of building server-side applications. Ideal for building web servers and APIs.
2. *EJS:* EJS (Embedded Javascript) is a templating language used to generate HTML markup with plain javascript. It allows embedding Javascript directly within HTML making the web application dynamic in nature.
3. *MongoDB:* A document based NoSQL database designed for flexibility, scalability and high performance. It is well-suited for handling large volumes of unstructured data.
4. *CSS:* CSS is used for styling a web application.
5. *spaCy:* open-source library for natural language processing in Python.
6. *Javascript:* High level programming language commonly used for making web applications interactive.
7. *scikit-Learn :* open source Python library for machine learning for building models.
8. *Flask:* web framework for Python for building web applications, APIs and microservices

B. *Major Modules*

This section comprises the modules included in the Digital Guide web application.

1. *Registration Module* users to create an account on the web application Digital Guide according to their requirements of what the user wants to learn in computer science engineering.
2. *Chatbot Module* users can communicate with the chatbot for emotional support, in stress time and for motivation
3. *Dashboard Module* provides users with personalized content according to their requirements. Provides users with the core functionalities of the web application such as video content, note making, chatbot, text summarizer and video summarizer.
4. *Video Summarizer Module* offers the users to paste the video link into the input field which provides a summary of the video, helping the user to get a quick revision of a video content.
5. *Text Summarizer Module* offers the users to provide a text content to get it summarized for a quick revision of notes.
6. *Progress Tracker Module* helps the user to track their progress once they complete the video content by clicking the status checkbox.
7. *Note Making Module* enables users to save notes while watching a video content to note down important key points while watching a video content on the web application.
8. *Profile Updation Module* provides users to update their personal information and update to some other learning content they want to learn.

C. Json Data Structure

The List data structure is used for storing video links for various domains and topics. The data structure consists of links containing the following information:

1. *Days:* The number of days to complete a video
2. *Topic:* Topic of the video content
3. *article:* reference article according to the video content.
4. practice: questions to practice according to the video content.

File Format:

[{

day: “21 to 22”, videolink: “https://www.youtube.com/embed/c2M-rlkkT5o” , topic: “learn mongodb”, article: “www.w3school.com/mongodb” , practice : “www.w3resource.com/mongodb-exercise” }]

D. BSON and CSV Data Structure

MongoDB uses BSON format to store user related information on MongoDB Atlas Cloud.

CSV is used to store data set for training machine learning model for chatbot

E. Chatbot working

This Flask web application creates a simple chatbot that uses machine learning to respond to user input. It leverages a Naive Bayes classifier with a TF-IDF vectorizer to process and classify text messages. The model is trained on a dataset of input messages and their corresponding responses. When a user submits a message through the web interface, the chatbot predicts a suitable response and displays it on the page.

The algorithm consists of the following steps:

Step 1: Load the data from the mental_health_chatbot.csv file. Separate the data into two parts X: Contains the input messages

, y: Contains the corresponding responses. # Load the dataset

data = pd.read_csv('mental_health_chatbot.csv', encoding='latin1')

Separate the data into input messages (X) and responses (y) X = data['Input Message']

y = data['Response']

Step 2: Preprocesses data using a TF-IDF Vectorizer to transform text into features.TF-IDF (Term Frequency-Inverse Document Frequency) gives more importance to words that are unique or rare within the dataset, helping the model focus on significant words.

from sklearn.feature_extraction.text import TfidfVectorizer

Initialize the TF-IDF Vectorizer vectorizer = TfidfVectorizer()

Fit and transform the input messages (X) to get the TF-IDF features

X_tfidf = vectorizer.fit_transform(X)

Step 3: Create and Train ModelSet up a pipeline combining the TF-IDF Vectorizer and a Naive Bayes Classifier.

Step 4: Set Up Flask App. Initialize a Flask app to handle HTTP requests and display the web interface.

Step 5: *Define the Home RouteCreate a route for the home page (/) that loads the index.html file. This file serves as the user interface for interacting with the chatbot.*

Step 6: Define the Ask RouteCreate a route (/ask) to handle POST requests with the user's input message.

Step 7: Run the Application. Start the Flask app in debug mode so it can display any errors for easy troubleshooting during development

F. Text Summarizer

spaCy, a powerful natural language processing (NLP) library, to generate a summary of a given text. It extracts key sentences from the input text by analyzing word frequency and sentence importance based on specific parts of speech, such as adjectives, proper nouns, verbs, and nouns. The script processes the text, scores each sentence based on the occurrence of important words, and selects the top sentences to form a concise, point-wise summary. The user can input the text directly as a command-line argument, and the program will return the summarized version with the most relevant points.

The algorithm consists of the following steps:

Step 1: Ask the user to enter the text they want summarized.

Step 2: Parse the article to extract the text and proceed to tokenize it.

Step 3: Using spacy model tokenize the individual words

,remove the common words and punctuation

Step 4: Count occurrences of each word using the counter model and normalize these word frequencies by dividing each by the maximum frequency.

Step 5: Assign each sentence a score based on the sum of its word frequencies ,scores of words are based on frequency of words.

Step 6: Choose the top sentences to display based on sentence scores ,which will be included in the final summary.

Step 7: Output the selected sentences in a point-wise format

Step 8: Display the final summary to the user in the terminal interface.

V. TESTING AND EVALUATION

A. *Testing methodologies*

1. Unit Testing:

- *Objective:* Validate the correctness and robustness of individual components, ensuring that each module functions as expected in isolation.
- *Approach:* Develop test cases for core functionalities, such as user registration, API communication, content retrieval, and NLP-based processing. Focus on text summarization accuracy and conciseness,
- *Expected Outcome:* Modules operate independently without errors, producing accurate outputs for a diverse range of inputs.

2. Integration Testing:

- *Objective:* Validate smooth interactions and data consistency across interconnected modules.
- *Approach:* Simulate real-world workflows, testing module interactions such as:
  - Passing data from user queries to the NLP models for summarization or ChatBot processing.
  - Synchronization between progress tracking, resource access, and backend storage.
- *Expected Outcome:* All components interact seamlessly, ensuring end-to-end functionality without data corruption, errors, or communication breakdowns.

3. System Testing:

- *Objective:* Verify the functionality and performance of the entire system in realistic, end-to-end scenarios.
- *Approach:* Conduct end-to-end testing for workflows such as summarizing youtube video from URL input to present an accurate textual summary.
- *Expected Outcome:* The system fulfills all specified requirements, providing an intuitive, secure, and reliable user experience.

B. *User feedback*

- **Objective:** Gather actionable feedback from target users to evaluate usability, functionality, and overall user satisfaction..
- **Methodology:**
  - Conduct trials involving end-users, including students, educators, and counselors..
  - Testing the Subject Guidance System's course recommendations.
  - Assessing the ChatBot's relevance and empathy in handling mental health queries.
- **Expected Outcome:** Insights for refining accuracy of text summarizer and youtube video summarizer, and improving interface on various kinds of devices.

C. *Performance metrics*

1. Response Time:

- *Metric:* Measure the accuracy of text summarization and ChatBot responses.
- *Expected Standard:* Text summaries achieve over 85% relevance to the original content and ChatBot responses are empathetic and contextually correct in over 90% of cases.

1. Response Time:

- *Metric:* Time required to process requests, such as summarization, course recommendation, or ChatBot queries.
- *Approach:* Load testing with simulated user volumes.
- *Expected Standard:* Response times remain under 1.5 seconds for text summarization, under 2 seconds for video processing, and under 1 second for chatbot responses.

VI. RESULTS

While the system is still under development, the expected results from testing and evaluation once the app is operational are:

1. **Distraction Free Environment:** Each individual module helps the user to learn through the video content in a distraction free environment as per their requirement.
2. **Study Material access:** Aggregate and display study resources from various domains, including textbooks, notes, videos, and articles.
3. **Video and Text summarization:** The video and text summarizer helps in a quick revision of a long video content and the later helps in getting a jist of long notes and article.

VII. CONCLUSION

The platform digital guide presents innovative tools and systems aimed at enhancing productivity, and mental well-being through the integration of **Machine learning** and **Natural Language Processing** (NLP).

The Text Summariser efficiently condenses textual information, making it more accessible to users. Similarly, the YouTube Video Summariser leverages advanced APIs and NLP techniques to generate concise video summaries, enabling quicker comprehension of visual content.The Subject Guidance System empowers students to explore and undertake courses in computer science tailored to their academic interests and career aspirations, while also providing a curated repository of study materials and progress tracking features.

. Additionally, the Chatbot offers a supportive platform for students to discuss mental health concerns, demonstrating the potential of NLP.

Together, these systems highlight the transformative potential of AI-driven technologies in addressing diverse challenges in education, content accessibility, and mental health support.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to Prof. Ashwini T N, Assistant Professor in the Department of Computer Science and Engineering at Bangalore Institute of Technology, for their invaluable guidance, unwavering support, and mentorship throughout this research. Their expertise and thoughtful suggestions played a crucial role in shaping the project and improving its overall quality.

Lastly, we express our deep appreciation to our family and friends for their constant encouragement and support throughout this research journey.

REFERENCES

[1] Picchetola M, Dieckbuch-Helmfeld L, "Enactivism and Digital Learning Platforms." (2020)

[2] Randhavane Y M, Gaikwad A B, "Intelligent Marathi Language Learning Platform with Adaptive Feedback." (2024)

[3] Ayachi Z, "Platform-driven English Language Learning." (2023)

[4] Xu L, Yu W, "Design and Implementation of Artificial Intelligence Online Learning Platform Based on Resource Scheduling Technology." (2024)

[5] Ruiz F A Z, Cevalón R M, Rojas Rivera W F, Mayhua J A C, "Khan Academy Platform for Mathematics Teaching and Learning." (2024)

[6] Karkar A J M, Fatlawi H K, Al-Jobouri A A, "Highlighting E-Learning Adoption Challenges Using Data Analysis Techniques: University of Kufa as a Case Study." (2020)

[7] Eduandze J K, Barf K A, "Perceptions of Senior High School Students towards E-Learning Platform in Some Selected Senior High Schools in Cape Coast Metropoli." (2015)

[8] Christian M, Maphalala O, Adigim T, "Academics' Experience of Implementing E-Learning in A South African Higher Education Institution." (2020)

[9] Duri S, Sara A, Lajane B F, Halawa, "Evaluation and Redesign of an E-Learning Platform." (2024)

[10] Khalek T A, Al-Hussain, "The Impact of Machine Learning Technology on E-Learning Platforms on Student Learning Behavior." (2024)