IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, March 2026
Impact Factor: 8.2

VidSage: AI-Powered Interactive Learning Platform Using YouTube Content The system utilizes OpenRouter API for AI-based Responses

Yash Patil, Rushikesh Surwase, Vivek Suryavanshi
Student, Information Technology
, Vishweshwarayya Institute of Engineering and Technology, Almala, India

Abstract: *This paper presents the design and implementation of VidSage, an AI-powered interactive learning platform: a Smart Learnering Management System with Real-Time Order Tracking, developed to digitize and modernize traditional learnering operations. Conventional learnering systems rely on manual record-keeping, which leads to issues such as data loss, inefficient order tracking, and poor customer communication. The system utilizes OpenRouter API for AI-based responses.*

The proposed system is built using the modern full-stack architecture (Next.js, Node.js, MongoDB, Socket.IO) and introduces a dual-role platform for both customers and learners. It enables digital order management, multi-item video handling with flexible measurement inputs, and portfolio-based learner discovery using geo-location features. A key contribution of this system is its offline-first architecture, where order data is temporarily stored in local storage when the backend server is unavailable and automatically synchronized once connectivity is restored. This ensures uninterrupted workflow, particularly in environments with unstable network conditions or server downtime.

The system also integrates features such as JWT-based authentication for secure access, Cloudinary-based image handling for portfolio management, and automated WhatsApp communication for order updates and invoice sharing. The implementation demonstrates improved efficiency in order handling, enhanced customer experience through real-time tracking, and reduced dependency on manual processes.

This solution provides a scalable and practical approach for small and medium-scale learnering businesses to transition into a digital ecosystem while maintaining operational flexibility..

Keywords: Learnering Management System, MERN Stack, Order Tracking System, JWT Authentication, MongoDB, React.js, Cloudinary Integration, Geo-Location Search.

I. INTRODUCTION

The Learnering Industry, Particularly In Small And Medium-Scale Businesses, Continues To Rely On Traditional Methods Such As Manual Record-Keeping Using Physical Registers, Commonly Known As “Learner Books.” While This Approach Has Been Followed For Decades, It Introduces Several Inefficiencies Including Difficulty In Managing Customer Data, Lack Of Proper Order Tracking, Risk Of Data Loss, And Limited Communication Between Learners And Customers. Additionally, Customers Often Face Challenges In Discovering Skilled Learners Based On Specific Requirements Such As Video Type Or Specialization, Leading To A Time-Consuming And Inconvenient Experience.

With The Advancement Of Digital Technologies, There Is A Growing Need To Transform These Manual Processes Into Efficient And Scalable Digital Solutions. Existing Systems In The Market Primarily Focus On Either E-Commerce Clothing Platforms Or Generic Service Listing Platforms, But They Fail To Address The Unique Workflow Of Learnering Businesses, Such As Handling Customized Measurements, Managing Multiple Videos Within A Single Order, And Maintaining Long-Term Customer Records.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, March 2026 Impact Factor: 8.2

To Address These Challenges, This Paper Proposes VidSage, A Smart Learnering Management System Developed Using The Mern Stack (Mongodb, Express.Js, React.Js, And Node.Js). The System Provides A Dual-Role Platform For Both Customers And Learners, Enabling Digital Order Management, Multi-Item Video Handling, And Real-Time Order Tracking. It Also Includes Portfolio-Based Learner Discovery Using Geo-Location Features, Allowing Customers To Find Suitable Learners Based On Their Needs And Location.

Furthermore, The System Incorporates Secure Authentication Using Json Web Tokens (Jwt), Flexible Database Design For Dynamic Measurement Storage, And Integration With Communication Platforms Such As Whatsapp For Order Updates And Invoice Sharing. By Digitizing Traditional Learnering Workflows, The Proposed System Improves Operational Efficiency, Enhances Customer Experience, And Reduces Dependency On Manual Processes.

II. LITERATURE REVIEW

The Digital Transformation Of Service-Based Industries Has Led To The Development Of Various Platforms For Online Booking, Service Discovery, And Customer Management. Several Existing Systems Attempt To Address Problems Related To Service Accessibility And Management, But They Fall Short When Applied To The Specific Needs Of The Learnering Industry.

Platforms Such As Online Marketplaces And Service Listing Applications (E.G., Real Estate And Local Service Platforms) Provide Features Like User Registration, Search Filters, And Basic Profile Management. These Systems Allow Users To Discover Service Providers Based On Categories And Location. However, They Are Primarily Designed For Standardized Services And Do Not Support Highly Personalized Workflows Such As Custom Video Stitching, Where Each Order May Involve Unique Measurements And Multiple Items.

Some Learnering Management Solutions And Small-Scale Applications Focus On Digitizing Customer Records And Order Details. While These Systems Improve Data Storage Compared To Manual Registers, They Often Lack Advanced Features Such As Multi-Item Order Handling, Structured Order Tracking, And Integrated Communication Mechanisms. Additionally, Many Of These Systems Do Not Provide A Unified Platform That Connects Both Customers And Learners, Limiting Their Usability In Real-World Scenarios.

Existing E-Commerce Platforms Offer Order Tracking And Payment Systems, But They Are Designed For Ready-Made Products Rather Than Customized Services. They Do Not Support Flexible Measurement Inputs Or Personalized Learnering Workflows. Similarly, Generic Customer Relationship Management (CRM) Systems Provide Tools For Managing Customer Data But Are Not Optimized For Domain-Specific Requirements Such As Video Measurement Variations And Learnering-Specific Order Processes.

Based On The Analysis Of Existing Systems, It Is Evident That There Is A Gap In Providing A Comprehensive Digital Solution Learnered Specifically For The Learnering Industry. The Proposed System, VidSage, Addresses These Limitations By Integrating Features Such As Dynamic Measurement Handling, Multi-Item Order Management, Real-Time Order Tracking, And Portfolio-Based Service Discovery Within A Single Platform. This Makes It More Suitable For Real-World Learnering Operations Compared To Existing Generalized Solutions.

III. METHODOLOGY

3.1 Development Methodology

VidSage was developed using an Agile-inspired iterative approach with two-week sprints, allowing for continuous feedback integration and course correction. The team adopted Kanban-style task management, test-driven development practices for critical features, and pair programming for complex architectural decisions. This methodology proved effective for managing the complexities of full-stack development, deployment challenges, and real-time system coordination.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, March 2026
Impact Factor: 8.2

3.2 System Architecture and Design Patterns

VidSage employs a decoupled microservices-inspired architecture separating frontend and backend concerns. The system implements the Repository Pattern for data access, the Strategy Pattern for pluggable AI providers, and Job Queue Pattern for asynchronous processing. This architecture enables independent scaling, facilitates testing, and maintains clear separation of concerns. Real-time communication uses the Namespace/Room pattern in Socket.IO to isolate collaborative sessions by jobId, preventing message leakage between concurrent users.

The backend further integrates multiple processing layers, including external CLI-based tools such as yt-dlp for audio extraction, ffmpeg for audio segmentation, and Whisper.cpp for offline transcription. These components work together in a pipeline model where each stage feeds into the next, ensuring efficient processing. Additionally, an in-memory job store is used to track the status, progress, and transcript data of each video processing request, enabling real-time feedback to the frontend.

3.3 Technology Stack Rationale

Next.js 14: Server-side rendering, API routes, and automatic code splitting for optimal frontend performance
React 19: Modern hooks, concurrent features, and ecosystem maturity for component reusability
Node.js + Express.js: Non-blocking I/O, massive npm ecosystem, and proven real-time capabilities
MongoDB Atlas: Flexible schema, scalability, and easy cloud hosting without infrastructure overhead
Socket.IO: Battle-tested real-time communication library with fallback mechanisms and automatic reconnection
OpenAI API: State-of-the-art language models with proven reliability and commercial support
yt-dlp: Maintained YouTube extraction tool with active development and robust error handling

3.4 Data Flow and Processing Pipeline

When a user submits a YouTube URL:
The backend validates the URL and creates a Job Queue entry
yt-dlp extracts audio asynchronously
Whisper CLI transcribes audio to text
OpenAI processes the transcript via RAG-enriched prompts to generate summaries and quiz content
The frontend polls the backend for job status updates
Once complete, the interactive dashboard loads with all generated content
Socket.IO establishes a real-time connection for collaborative study rooms
This pipeline ensures non-blocking user experience with clear progress feedback.

To ensure efficient processing, the system uses a non-blocking job execution model. Once a job is initiated, the backend immediately responds with a unique job identifier while continuing processing in the background. The frontend periodically polls the backend to retrieve progress updates. This approach improves user experience by avoiding long wait times and enabling real-time visibility into processing stages. The segmented transcription approach also allows partial transcripts to be displayed before full completion.

IV. IMPLEMENTATION

4.1 Frontend Architecture

The frontend employs a component-based architecture with React hooks for state management. Key components include VideoPlayer (HTML5 video with custom controls), AITutor (real-time chat interface), TranscriptPanel (searchable, synchronized transcript), QuizModule (interactive multiple-choice assessment), StudyRoom (real-time collaborative space), and NotesPanel (persistent note management). Tailwind CSS provides responsive styling, while Lucide React supplies consistent iconography.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, March 2026
Impact Factor: 8.2

4.2 Backend Architecture and Job Queue System

The backend implements a robust job queue system utilizing in-memory queuing with graceful persistence. When a URL is submitted: (1) A Job object is created with status 'processing', (2) yt-dlp extraction begins asynchronously, (3) Whisper transcription occurs in sequence, (4) OpenAI processes content concurrently with P-Limit for concurrency control, (5) Results are stored in server-side temporary memory, (6) Frontend continuously polls GET /api/jobs/:jobId until completion. This design ensures the API remains responsive and frontend receives real-time updates without blocking.

The backend follows a modular service-oriented architecture where responsibilities are divided into controllers, services, and middleware. Controllers handle incoming requests, services contain business logic such as audio processing and AI interaction, and middleware ensures authentication and request validation.

Audio processing is handled through a pipeline involving yt-dlp for downloading, ffmpeg for segmentation, and Whisper.cpp for transcription. The use of concurrency control via libraries such as p-limit ensures optimal CPU utilization without overloading the system. Each transcription segment is processed sequentially or in controlled parallelism, and results are appended incrementally.

The system also uses in-memory data stores for managing job states and AI session histories. While this approach simplifies development and improves performance, it also means that data is not persistent across server restarts, which is considered an acceptable trade-off for the current implementation.

4.3 AI Integration and RAG Implementation

The AI Tutor leverages OpenAI's GPT models with a Retrieval-Augmented Generation (RAG) pattern. Instead of providing raw transcripts, the system creates structured context by extracting key sections relevant to user questions. When a user asks a question in the AI Tutor, the system: (1) Searches the transcript for semantically similar segments, (2) Constructs a prompt including the question, relevant transcript sections, and system instructions, (3) Sends to OpenAI API, (4) Returns the response with citation information. This approach maintains accuracy while reducing hallucinations and token costs.

Instead of directly relying on proprietary AI APIs, the system uses the OpenRouter API as a unified interface for accessing multiple large language models. This approach provides flexibility in model selection and cost optimization. The default model used is Mistral 7B, which offers a balance between performance and efficiency.

The AI system operates using a context-driven approach, where relevant portions of the transcript are combined with the user's query to form a structured prompt. Additionally, recent conversation history is included to maintain context continuity. This ensures that responses are accurate, relevant, and conversational in nature.

The system also limits the number of past messages included in each request to avoid exceeding token limits and to maintain performance efficiency.

4.4 Real-Time Collaboration via Socket.IO

Study Rooms implement Socket.IO namespaces and rooms for secure, isolated collaboration. Each job session creates a unique namespace /vidsage/:jobId. When users join: (1) They are added to a room specific to that jobId, (2) Messages broadcast only within that room (isolation), (3) User presence is tracked with join/leave events, (4) Typing indicators and message timestamps provide rich real-time feedback, (5) Socket disconnection handling with reconnect logic ensures resilience.

The real-time communication system is designed using a room-based architecture, where each study session is associated with a unique jobId. This ensures that messages are isolated within specific groups and prevents cross-interference between different sessions. The use of WebSockets enables low-latency communication, making the interaction feel instantaneous for users.

Authentication for socket connections is handled using the same JWT mechanism as HTTP requests, ensuring consistent security across communication channels.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, March 2026
Impact Factor: 8.2

4.5 Deployment and Infrastructure

Frontend: Deployed on Railway using Next.js 14 build output. Custom domain vidsage.in routes through Cloudflare with Full SSL mode. Environment variables securely injected for API endpoints.

Backend: Deployed on AWS EC2 (Windows instance) running Node.js with yt-dlp, ffmpeg, and Deno installed via custom startup scripts. PM2 manages process execution and auto-restart. API endpoint api.vidsage.in uses Cloudflare DNS pointing to EC2 elastic IP with Flexible SSL mode.

V. RESULTS AND DISCUSSION

The implementation of VidSage was tested across multiple scenarios to evaluate its functionality, usability, and performance in real-world learnering operations. The system was successfully able to handle core operations such as user authentication, order creation, multi-item video management, measurement storage, and real-time order tracking.

Functional Results

The system was tested with different user roles, including both customers and learners.

Learners were able to:

Create and manage orders efficiently

Store and retrieve customer measurements dynamically

Update order status at different stages

Generate WhatsApp messages for customer communication

Customers were able to:

Search for learners based on specialization and location

View learner portfolios

Track order progress through different stages

These functionalities confirm that the system meets its primary objective of digitizing traditional learnering workflows.

System Performance

The application demonstrated smooth performance under normal usage conditions:

Fast UI rendering due to React.js component-based architecture

Efficient API response handling using Node.js and Express.js

Quick data retrieval from MongoDB due to flexible schema design

The system performs well for small to medium-scale usage, making it suitable for local learnering businesses.

Usability and User Experience

The user interface was designed to be simple and intuitive:

Dashboard-based navigation allows learners to manage orders easily

Clear order status indicators improve tracking visibility

Minimal learning curve for users unfamiliar with digital systems

The integration of WhatsApp messaging further enhances user convenience by simplifying communication between learners and customers.

VI. DISCUSSION

The results indicate that the proposed system effectively addresses the limitations of traditional manual record-keeping. By digitizing order management and customer data, the system reduces the chances of data loss and improves operational efficiency.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, March 2026 Impact Factor: 8.2

The use of MongoDB enables flexible handling of varying measurement data, which is a critical requirement in learnering systems. Additionally, the modular architecture ensures that the system can be extended with new features such as online payments or mobile application support.

However, the system is currently optimized for small-scale deployment and may require further enhancements such as scalability optimization and advanced analytics for handling large-scale operations.

VII. CONCLUSION

The development of VidSage demonstrates an effective approach to digitizing traditional learnering operations through a modern web-based system. The project successfully addresses key challenges associated with manual record-keeping, such as inefficient order management, lack of tracking, and difficulty in maintaining customer data.

By implementing a MERN stack-based architecture, the system provides a scalable and efficient platform for both learners and customers. Features such as multi-item order management, dynamic measurement handling, real-time order tracking, and integrated communication through WhatsApp significantly improve the overall workflow and user experience. The system proves to be a practical solution for small and medium-scale learnering businesses, enabling them to transition from manual processes to a structured digital environment. It enhances operational efficiency, reduces errors, and simplifies customer interaction.

In conclusion, the project not only fulfills its intended objectives but also establishes a strong foundation for future enhancements such as mobile application development, online payment integration, and advanced analytics. This makes VidSage a scalable and impactful solution in the domain of digital learnering management systems.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my project guide for their valuable guidance, continuous support, and encouragement throughout the development of this project. Their insights and suggestions have been instrumental in shaping this work.

I am also thankful to the Head of the Department and all the faculty members of the Computer Engineering Department for providing the necessary resources and a supportive environment to complete this project successfully.

I extend my appreciation to my teammates for their cooperation, collaboration, and contribution during the project development process.

Finally, I would like to thank my family and friends for their constant support and motivation, which helped me complete this project.

REFERENCES

[1]. MongoDB Documentation, *"MongoDB Manual"*. Available: https://www.mongodb.com/docs/

[2]. Express.js Documentation, *"Fast, unopinionated, minimalist web framework for Node.js"*. Available: https://expressjs.com/

[3]. React.js Documentation, *"React – A JavaScript library for building user interfaces"*. Available: https://reactjs.org/

[4]. Node.js Documentation, *"Node.js Official Documentation"*. Available: https://nodejs.org/

[5]. Mongoose Documentation, *"MongoDB Object Modeling for Node.js"*. Available: https://mongoosejs.com/

[6]. Cloudinary Documentation, *"Cloudinary Image and Video Management"*. Available: https://cloudinary.com/documentation

[7]. JSON Web Token (JWT), *"JWT Introduction"*. Available: https://jwt.io/introduction

[8]. Axios Documentation, *"Promise-based HTTP client for the browser and Node.js"*. Available: https://axios-http.com/

[9]. Bcrypt Documentation, *"A library to hash passwords"*. Available: https://www.npmjs.com/package/bcrypt

[10]. WhatsApp Business API, *"Click to Chat Feature"*. Available: https://faq.whatsapp.com/