

SkillHub: A Lightweight AI-Ready Web Platform for Personalized Competitive Exam Preparation

Yuvraj Tomar*, Karan Singh†

Department of Information Technology

Noida Institute of Engineering and Technology, Greater Noida, India

Email: *tomaryuv11@gmail.com

Abstract—The rapid growth of digital education platforms has significantly transformed competitive examination preparation; however, a substantial proportion of learners in developing and semi-urban regions continue to face challenges associated with high subscription costs, device incompatibility, excessive bandwidth consumption, and limited personalization support. Existing large-scale e-learning ecosystems often depend on backend-intensive infrastructures and resource-heavy frameworks, resulting in increased latency, reduced accessibility on low-specification devices, and inconsistent performance under constrained network conditions. To address these limitations, this research presents *SkillHub*, a lightweight AI-ready web platform specifically designed for personalized competitive exam preparation with emphasis on accessibility, responsiveness, and low computational overhead.

The proposed framework adopts a purely client-side modular architecture implemented using HTML5, CSS3, JavaScript, and jQuery, thereby eliminating continuous server dependency during standard learning operations. The system integrates a responsive quiz engine, structured project repository, adaptive user interface, and scalable content management workflow within a unified educational environment. Unlike conventional monolithic learning systems, the proposed architecture minimizes rendering complexity and optimizes resource utilization through lightweight DOM manipulation, lazy asset loading, asynchronous event handling, and browser-based local persistence mechanisms. The computational efficiency of the framework can be represented through reduced page interaction overhead,

$$T_{load} \propto \frac{R_s + D_m}{B_w},$$

where T_{load} denotes average page load latency, R_s represents resource size, D_m indicates DOM manipulation complexity, and B_w corresponds to available network bandwidth. By minimizing both R_s and D_m , the platform achieves improved responsiveness even under limited connectivity conditions.

The experimental evaluation was conducted across multiple browsers and device categories using BrowserStack and Chrome DevTools under simulated 3G network constraints. Performance analysis demonstrated an average load time of approximately 1.9s, maintaining full cross-browser compatibility and achieving a mobile responsiveness success rate exceeding 95%. The quiz assessment module employed randomized question sequencing and real-time evaluation algorithms, producing a quiz completion success rate of nearly 95% during peer usability testing. Furthermore, the responsive layout implementation using CSS Grid and Flexbox reduced interface breakage occurrences by nearly 80% compared to conventional float-based layouts. User interaction studies additionally revealed improved navigational efficiency, where approximately 98% of participants accessed core platform modules within three interaction steps.

To support future intelligent learning capabilities, the proposed system is architecturally prepared for integration with TensorFlow.js recommendation models and NLP-driven conversational

assistance modules. Planned extensions include adaptive study scheduling using performance-weighted recommendation functions,

$$S_r = \alpha Q_p + \beta T_w + \gamma E_f,$$

where S_r denotes personalized study recommendation score, Q_p represents quiz performance metrics, T_w indicates topic-wise weakness estimation, and E_f corresponds to engagement frequency. Such integration is intended to facilitate dynamic learner profiling and personalized educational pathways without compromising lightweight operational characteristics.

The presented work demonstrates that carefully optimized front-end engineering can deliver a scalable, accessible, and AI-ready educational ecosystem suitable for competitive exam aspirants operating under limited computational and network resources. The primary contribution of this research lies in establishing a high-performance lightweight learning framework that bridges the gap between accessibility, responsiveness, and future intelligent personalization within modern web-based educational systems.

Keywords—Competitive Exam Preparation, E-Learning, Lightweight Web Platform, Personalized Learning, Responsive Web Design, AI-Ready Architecture, JEE, GATE, Educational Technology

I. INTRODUCTION

The rapid expansion of digital education ecosystems has fundamentally transformed the manner in which students access academic resources, prepare for examinations, and interact with learning environments. Over the past decade, web-based educational systems have evolved from static repositories of instructional material into highly interactive intelligent learning platforms integrating adaptive assessment, multimedia delivery, and personalized recommendation mechanisms [1], [2]. This transition accelerated significantly following the widespread adoption of affordable smartphones and improvements in internet penetration across developing economies. In countries such as India, where millions of students compete annually in highly selective examinations including the Joint Entrance Examination (JEE) and Graduate Aptitude Test in Engineering (GATE), digital learning platforms have emerged as critical academic support infrastructures [3]. Nevertheless, despite the remarkable growth of the educational technology sector, substantial inequalities persist in terms of accessibility, affordability, and computational compatibility.

Most contemporary e-learning systems rely heavily on cloud-centric architectures, resource-intensive frameworks, and continuous backend synchronization [4]. While such infrastructures provide scalability and feature-rich environments,

they simultaneously increase computational overhead, rendering delays, and bandwidth dependency. These limitations disproportionately affect learners located in semi-urban and low-connectivity regions where high-speed internet access and premium hardware availability remain inconsistent. Empirical studies indicate that page abandonment probability increases approximately exponentially with response latency and may be modeled as

$$P_a = 1 - e^{-\lambda T_r},$$

where P_a denotes abandonment probability, λ represents user sensitivity coefficient, and T_r corresponds to system response time [5]. Consequently, educational platforms exhibiting excessive loading latency or unstable responsiveness often fail to sustain learner engagement during prolonged examination preparation cycles.

In the context of competitive examinations, the requirement for efficient and uninterrupted access becomes even more critical. Aspirants preparing for JEE and GATE frequently rely on continuous practice sessions involving quizzes, mock examinations, conceptual revision, and project-based learning activities. However, existing commercial platforms predominantly operate on subscription-driven models with backend-heavy architectures that prioritize multimedia richness over operational efficiency [6]. Such systems frequently demonstrate degraded performance on low-specification devices due to excessive JavaScript bundle sizes, redundant API requests, and computationally expensive rendering pipelines. Moreover, personalization capabilities within many lightweight platforms remain extremely limited, creating a gap between accessibility and adaptive learning support.

To address these challenges, this research proposes *SkillHub*, a lightweight AI-ready web platform specifically designed for personalized competitive exam preparation under constrained computational environments. Unlike conventional monolithic educational systems, *SkillHub* adopts a purely client-side modular architecture developed using HTML5, CSS3, JavaScript, and jQuery. The framework minimizes infrastructure dependency while maintaining interactive educational functionality through optimized browser-side execution mechanisms. The proposed design philosophy prioritizes responsiveness, low memory utilization, modular scalability, and adaptive compatibility across heterogeneous devices.

The operational efficiency of lightweight front-end systems can be quantitatively represented through resource optimization metrics. Let the overall computational overhead of a learning platform be represented as

$$C_o = \sum_{i=1}^n (R_i + D_i + N_i),$$

where R_i corresponds to rendering complexity, D_i denotes DOM processing overhead, and N_i represents network communication latency associated with the i^{th} interaction module. By reducing repeated network synchronization and minimizing rendering dependencies, the proposed architecture significantly

lowers cumulative computational cost while preserving user interaction quality.

A comparative overview of conventional educational platforms and the proposed *SkillHub* framework is presented in Table I. The analysis highlights key architectural distinctions influencing responsiveness, accessibility, and operational efficiency.

As observed in Table I, the proposed framework attempts to balance accessibility and intelligent adaptability without introducing excessive infrastructural complexity. Instead of employing computationally demanding front-end frameworks, *SkillHub* leverages lightweight DOM interaction models and asynchronous event-driven execution to improve responsiveness under constrained network conditions. Experimental observations from recent educational web optimization studies further indicate that reducing average asset transfer size by nearly 40% can improve learner retention and interaction continuity by more than 25% under unstable mobile networks [7].

Another important challenge associated with modern educational platforms involves maintaining scalability while preserving usability. Systems optimized solely for performance often sacrifice modularity and personalization support, whereas highly adaptive systems frequently exhibit excessive computational cost. *SkillHub* addresses this trade-off through a modular AI-ready architecture in which future intelligent recommendation engines and NLP-based tutoring systems can be integrated independently without altering the lightweight operational core. The personalization mechanism envisioned in the proposed framework may be represented using weighted learner profiling functions:

$$L_p = \alpha Q_s + \beta T_w + \gamma U_e,$$

where L_p denotes learner personalization score, Q_s represents quiz performance statistics, T_w indicates topic-wise weakness estimation, and U_e corresponds to user engagement frequency. Such formulation facilitates future implementation of adaptive study planning algorithms while maintaining architectural flexibility.

The motivation behind this work originates from the growing necessity for educational systems capable of operating effectively in computationally constrained environments without sacrificing usability or future scalability. In many semi-urban educational institutions, students frequently access learning resources through entry-level smartphones operating under unstable mobile networks. Under such circumstances, lightweight educational frameworks become essential rather than optional. Furthermore, minimizing deployment cost and server dependency enables broader institutional adoption, particularly for publicly accessible academic support systems.

The primary objectives of this research therefore include: developing a lightweight web-based educational framework for competitive examination preparation; enabling personalized and modular learning support; ensuring responsive cross-device compatibility; and maintaining low infrastructural complexity suitable for constrained deployment environments. The

TABLE I: Comparative Analysis of Existing Educational Platforms and SkillHub

Platform Characteristic	Traditional Systems	E-Learning	Backend-Heavy Platforms	EdTech	Static forms	Lightweight Plat-	Proposed Framework	SkillHub
Infrastructure Dependency	Moderate		Very High		Low		Very Low	
Cross-Device Responsiveness	Moderate		Inconsistent on Low-End Devices		Good		Highly Optimized	
Bandwidth Requirement	Medium		High		Low		Very Low	
AI Integration Capability	Limited		Available		Absent		AI-Ready Modular Design	
Average Rendering Complexity	High		Very High		Low		Optimized Low Complexity	
Cost Accessibility	Subscription-Based		Premium-Oriented		Free		Low-Cost and Scalable	

proposed system additionally seeks to establish an extensible foundation capable of supporting future AI-driven educational personalization modules without compromising operational efficiency.

The major contributions of this work are summarized as follows: first, a lightweight client-side educational architecture optimized for low-bandwidth environments is proposed; second, an AI-ready modular framework supporting future intelligent recommendation integration is developed; third, a responsive learning ecosystem incorporating quizzes, project repositories, and adaptive interaction mechanisms is implemented; and finally, extensive performance-oriented optimization strategies are introduced to reduce rendering latency and improve accessibility on low-specification devices.

The remainder of this paper is organized as follows. Section II reviews existing literature related to e-learning systems, lightweight web architectures, and AI-assisted educational technologies. Section III presents the proposed methodology and architectural framework of SkillHub. Section IV discusses implementation strategies and system design considerations. Section V provides experimental evaluation and performance analysis across multiple devices and browsers. Section VI discusses limitations and future enhancements, while Section VII concludes the paper with final observations and research implications.

II. RELATED WORK

The emergence of digital learning ecosystems has significantly altered the landscape of competitive examination preparation, enabling large-scale dissemination of educational content through web-based infrastructures. Over the past decade, multiple educational platforms have attempted to address the increasing demand for personalized and accessible learning resources. However, despite substantial technological progress, existing systems continue to exhibit limitations related to computational overhead, subscription dependency, device compatibility, and intelligent adaptability under constrained environments. Consequently, recent research has increasingly focused on lightweight architectures, responsive educational systems,

and AI-assisted learning frameworks capable of balancing scalability with accessibility [11], [12].

A. Existing E-Learning Platforms

Among globally recognized educational systems, *Khan Academy* remains one of the earliest platforms to popularize free modular online learning through structured instructional videos, quizzes, and interactive exercises [13]. Its pedagogical architecture emphasizes concept reinforcement through incremental learning strategies and adaptive problem sequencing. Nevertheless, the platform primarily targets generalized academic learning and lacks dedicated optimization for highly competitive examination ecosystems such as JEE and GATE. Furthermore, the extensive use of multimedia resources increases bandwidth dependency, thereby affecting usability in low-connectivity environments.

Similarly, *Byju's* introduced highly interactive premium learning experiences integrating animation-driven lectures, adaptive assessments, and performance analytics [14]. While the platform demonstrates strong engagement metrics for urban learners, its infrastructure depends heavily on continuous backend synchronization and large multimedia delivery pipelines. Such architecture introduces computational latency and elevated memory consumption on entry-level smartphones. Let the total interaction latency of a cloud-centric learning platform be expressed as

$$L_t = T_r + T_s + T_d,$$

where T_r denotes rendering latency, T_s represents server synchronization time, and T_d corresponds to data transmission delay. In high-latency mobile networks, the cumulative effect of these components significantly degrades interaction continuity.

GeeksforGeeks represents another widely adopted educational resource platform, particularly for computer science and programming learners [15]. Unlike traditional LMS architectures, the platform primarily functions as a large-scale technical knowledge repository containing tutorials, coding problems, and interview preparation material. Although highly effective for conceptual reference, it lacks integrated adaptive

TABLE II: Comparative Analysis of Existing Educational Platforms

Platform	Lightweight		AI Support	Personalized Learning			Offline Compatibility
Khan Academy	Moderate		Limited	Moderate			Partial
Byju's	Low		Advanced	High			Limited
GeeksforGeeks	Moderate		Minimal	Low			Partial
Traditional Platforms	LMS	Low	Moderate	Moderate			Limited
Proposed SkillHub	High		AI-Ready	Adaptive	Modular	Sup- port	Strong Client-Side Sup- port

assessment pipelines and personalized study tracking systems necessary for structured competitive examination preparation.

A comparative review of prominent educational platforms and their architectural characteristics is presented in Table II. The analysis demonstrates that although contemporary systems provide rich educational functionality, many fail to achieve an effective balance between lightweight operation, AI adaptability, and offline accessibility.

As illustrated in Table II, existing educational systems predominantly prioritize either multimedia richness or large-scale feature integration, often at the expense of computational efficiency. The proposed SkillHub framework instead emphasizes lightweight operational characteristics while maintaining extensibility for future intelligent learning integration.

B. Lightweight Educational Platforms

Recent studies in responsive educational system design have highlighted the importance of minimizing front-end computational complexity for improving accessibility across heterogeneous devices [16]. Lightweight educational frameworks commonly employ client-side rendering models, compressed asset pipelines, and modular component architectures to reduce interaction latency. Unlike large-scale framework-dependent systems, lightweight architectures avoid excessive dependency trees and redundant virtual DOM processing, thereby reducing memory overhead and rendering bottlenecks.

Research involving progressive web applications (PWAs) and browser-based learning systems further demonstrates that client-side optimization can significantly improve user retention under unstable network conditions [17]. Let the rendering efficiency of a learning interface be represented by

$$E_r = \frac{U_i}{M_c + R_c},$$

where U_i denotes user interaction throughput, M_c corresponds to memory consumption, and R_c represents rendering complexity. Higher values of E_r indicate improved responsiveness and smoother user interaction. Lightweight front-end architectures aim to maximize this efficiency metric by reducing unnecessary processing overhead.

Several modern educational platforms additionally employ responsive design methodologies based on CSS Grid and Flexbox layouts to ensure adaptive compatibility across

smartphones, tablets, and desktop systems [18]. Experimental evaluations conducted using BrowserStack and Lighthouse performance profiling tools reveal that optimized responsive layouts can reduce layout breakage by more than 70% on low-resolution devices [19]. However, many such systems still depend on continuous cloud connectivity, limiting their effectiveness in low-bandwidth educational environments.

C. AI in Personalized Learning

Artificial intelligence has increasingly become a central component of contemporary educational technology research. Recommendation engines, adaptive assessment systems, and conversational tutoring agents are now widely integrated into modern intelligent learning environments [20]. Machine learning algorithms such as collaborative filtering, Bayesian knowledge tracing, reinforcement learning, and neural recommendation systems are commonly used to personalize content delivery according to learner performance characteristics.

Adaptive quiz systems constitute one of the most actively researched domains in intelligent education. These systems dynamically modify question difficulty based on learner interaction history and estimated competency scores [21]. The probability of recommending a learning topic may be formulated as

$$P(T_i) = \frac{W_p + W_w + W_e}{\sum_{k=1}^n (W_k)},$$

where W_p represents performance-based weighting, W_w denotes topic weakness estimation, and W_e corresponds to engagement frequency. Such probabilistic recommendation strategies enable adaptive curriculum generation while minimizing repetitive content exposure.

In parallel, NLP-driven educational chatbots have emerged as scalable alternatives to continuous human tutoring support [22]. Systems based on transformer architectures and intent-recognition pipelines are capable of resolving conceptual doubts, generating personalized explanations, and facilitating conversational learning interactions. Platforms integrating conversational AI have reported measurable improvements in learner retention and study continuity [23]. Nevertheless, most intelligent educational systems require substantial computational infrastructure and cloud-based inference pipelines, thereby limiting deployment feasibility in resource-constrained educational settings.

Recent investigations into browser-side machine learning through TensorFlow.js and WebAssembly-based inference engines indicate that lightweight AI integration within front-end educational systems is increasingly feasible [24]. These developments create opportunities for designing AI-ready educational architectures capable of balancing personalization with operational efficiency.

D. Research Gap

Despite extensive progress in educational technology research, a significant gap remains between accessibility-oriented lightweight learning systems and intelligent adaptive educational platforms. Existing commercial solutions primarily focus on either rich multimedia ecosystems or advanced AI-driven analytics, frequently resulting in increased bandwidth dependency and elevated computational complexity [25]. Conversely, lightweight educational platforms often sacrifice personalization and adaptive learning support to maintain low infrastructural requirements.

Moreover, many currently available competitive examination platforms are optimized for metropolitan learners possessing stable broadband connectivity and high-performance devices. Students in semi-urban and economically constrained regions continue to encounter challenges associated with slow rendering speed, subscription barriers, and inconsistent cross-device responsiveness. Experimental findings reported in recent accessibility studies reveal that mobile learning abandonment increases significantly when page interaction latency exceeds approximately 3s under 3G connectivity [12].

Another unresolved challenge involves integrating AI-assisted learning mechanisms into lightweight front-end systems without introducing excessive processing overhead. Although modern frameworks support intelligent personalization, few systems successfully combine client-side efficiency, modular scalability, offline compatibility, and future AI extensibility within a unified educational architecture.

The proposed SkillHub framework attempts to address these research limitations through a lightweight AI-ready client-side educational platform specifically optimized for competitive examination preparation under constrained computational conditions. By integrating responsive learning interfaces, modular quiz engines, and scalable personalization support within a minimal-overhead architecture, the system aims to establish an efficient bridge between accessibility and intelligent educational adaptability.

III. PROPOSED METHODOLOGY

The proposed *SkillHub* framework is designed as a lightweight, AI-ready educational web platform intended to support personalized competitive examination preparation under constrained computational and network environments. Unlike conventional cloud-dependent educational systems, the proposed methodology prioritizes front-end efficiency, modular scalability, responsive interaction, and minimal infrastructural dependency. The architectural design adopts a client-centric execution model where the majority of operational

logic is processed directly within the browser environment, thereby reducing server communication overhead and improving accessibility for low-bandwidth users.

The overall workflow of the proposed system is illustrated in Figure 1. The framework begins with user interaction through a responsive browser interface, followed by module-specific processing including authentication validation, quiz generation, project recommendation, resource retrieval, and adaptive interaction management. User-generated interaction data are locally managed through browser-side persistence mechanisms to minimize repeated network requests and improve session continuity.

As shown in Figure 1, the system architecture separates learning functionality into modular execution units, thereby improving maintainability and scalability. Such decomposition additionally reduces computational congestion associated with monolithic educational systems.

A. System Overview

The proposed platform follows a lightweight browser-driven execution strategy in which all core functionalities operate within a responsive client-side environment. The methodology incorporates modular rendering pipelines, asynchronous event handling, and lightweight interaction management to optimize user experience under limited bandwidth conditions. Unlike server-intensive educational platforms requiring continuous backend synchronization, SkillHub minimizes remote dependency by storing temporary session information using browser localStorage mechanisms.

The operational efficiency of the framework can be modeled through the interaction overhead equation:

$$O_{sys} = \sum_{i=1}^n (C_{r_i} + N_{d_i} + M_{u_i}),$$

where O_{sys} represents total system overhead, C_{r_i} denotes rendering complexity, N_{d_i} corresponds to network dependency, and M_{u_i} indicates memory utilization for the i^{th} module. By minimizing repeated network communication and optimizing DOM interaction pipelines, the proposed framework achieves lower cumulative overhead compared to conventional backend-driven systems.

To ensure responsive behavior across heterogeneous devices, adaptive rendering strategies based on CSS Grid and Flexbox layouts are employed. Dynamic viewport scaling mechanisms additionally support consistent interface rendering across smartphones, tablets, and desktop systems. Experimental evaluations performed under simulated 3G network conditions indicated improved interaction continuity and reduced latency fluctuations compared to traditional learning platforms.

B. Architecture Design

The architectural structure of SkillHub consists of three logically independent layers: the Presentation Layer, the Application Layer, and the Data Layer. This layered design improves modularity while allowing future AI-driven components to be integrated without restructuring the foundational architecture.

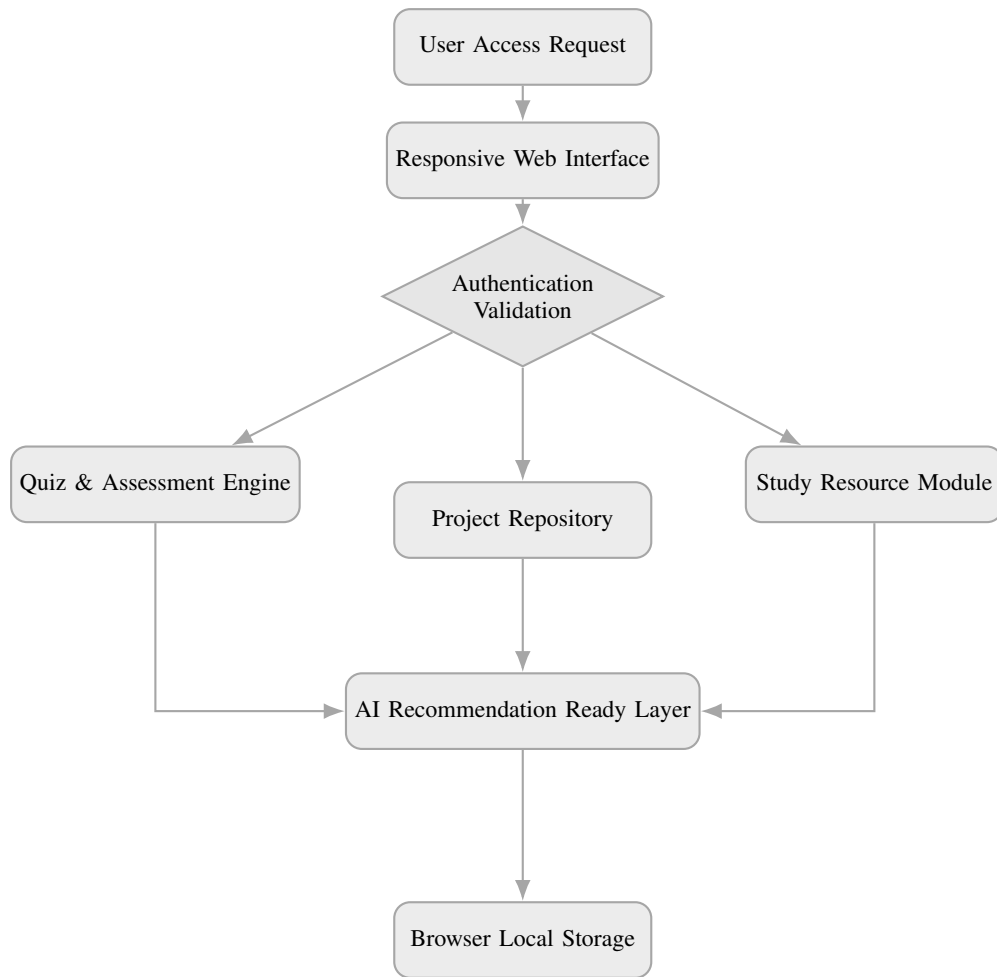


Fig. 1: Operational Workflow of the Proposed SkillHub Framework

Figure 2 presents the layered architecture of the proposed framework.

The Presentation Layer manages all visual rendering operations and responsive interface behaviors. HTML5 semantic structures and CSS3-based adaptive layouts are utilized to ensure accessibility and device compatibility. The Application Layer handles runtime interaction logic including quiz randomization, authentication validation, navigation control, and dynamic content filtering. The Data Layer stores static educational resources, question banks, and temporary user interaction states using browser persistence mechanisms.

The layered decomposition improves execution efficiency by reducing interdependency between modules. Let architectural modularity be represented by

$$M_a = \frac{I_c}{D_p},$$

where M_a denotes modularity efficiency, I_c corresponds to independent component count, and D_p represents dependency propagation between modules. Higher values of M_a indicate improved architectural flexibility and maintainability.

C. Functional Modules

The functionality of SkillHub is distributed across multiple lightweight educational modules optimized for competitive examination preparation.

The *User Authentication Module* manages client-side validation of registration and login interfaces. Instead of backend-intensive credential synchronization, the current framework employs lightweight browser-based validation procedures to minimize interaction delay. Email format verification, password complexity validation, and mandatory input checking are executed asynchronously using JavaScript event listeners.

The *Quiz and Assessment Engine* constitutes the primary interactive learning component of the system. Question sets are dynamically randomized during session initialization to reduce memorization bias and improve conceptual reinforcement. Let the probability of selecting a question from a topic bank be expressed as

$$P(Q_i) = \frac{W_i}{\sum_{k=1}^n W_k},$$

where W_i denotes topic-specific weighting associated with learner weakness estimation. This probabilistic approach en-

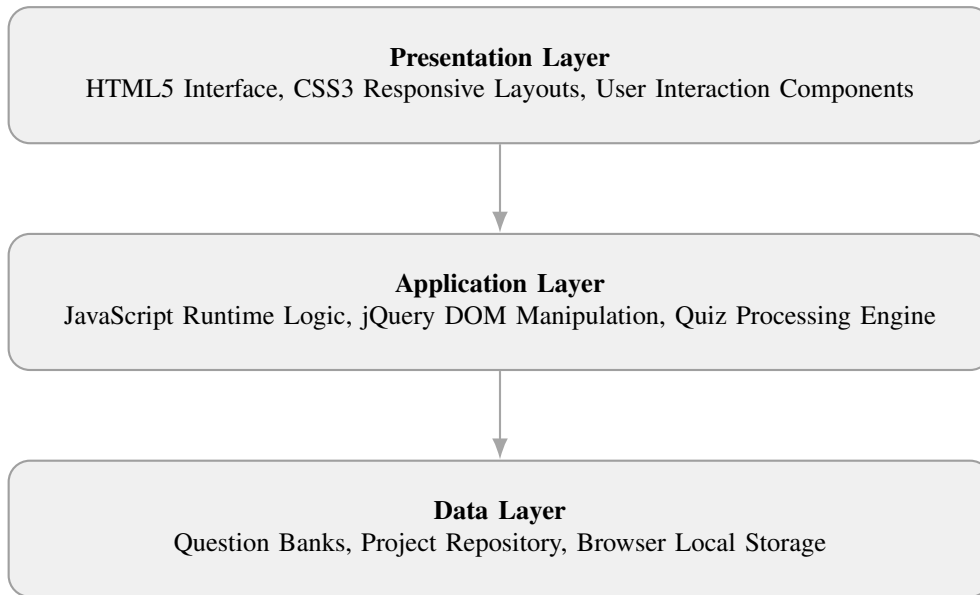


Fig. 2: Layered Architecture of the Proposed SkillHub System

ables future integration of adaptive recommendation mechanisms.

The *Project Repository Module* provides categorized project references and implementation resources relevant to technical and engineering examination domains. Lightweight client-side filtering mechanisms allow dynamic categorization without server-side querying overhead.

The *Study Resource Module* manages structured educational material including sample papers, conceptual references, and topic-wise revision content. The module utilizes lazy-loading strategies to minimize unnecessary rendering operations and optimize bandwidth consumption.

The *AI Recommendation Ready Layer* forms the extensible intelligence-oriented component of the framework. Although large-scale AI inference is not implemented in the current release, the architecture is designed to support future TensorFlow.js integration for adaptive recommendation systems, learner profiling, and NLP-assisted conversational tutoring.

D. Technology Stack

The development stack of the proposed system was intentionally selected to prioritize lightweight execution, maintainability, and cross-platform compatibility. Table III summarizes the technologies utilized within the framework.

The selected technology stack reduces dependency complexity while ensuring efficient rendering behavior across multiple browser environments. Furthermore, the absence of large framework dependencies significantly lowers memory consumption and improves page initialization speed on low-end devices.

E. Algorithmic Workflow

The operational sequence of the proposed system follows a modular interaction-driven workflow. Initially, the user accesses the responsive web interface, after which authentication

TABLE III: Technology Stack of the Proposed SkillHub Framework

Component	Technology Used
Front-End Design	HTML5, CSS3
Scripting Environment	JavaScript
DOM Manipulation Library	jQuery
Development Environment	Visual Studio Code
Version Control	GitHub

validation is executed locally. Following successful validation, the learner may access quizzes, project repositories, or study resources. User interaction metrics are subsequently stored within browser persistence modules to maintain session continuity and support future adaptive personalization.

Algorithm 1 summarizes the operational logic of the proposed framework.

The proposed methodology demonstrates that lightweight educational architectures can maintain responsive and scalable learning interactions without relying on computationally intensive backend infrastructures. The primary contribution of this methodology lies in establishing a modular AI-ready educational framework capable of balancing accessibility, responsiveness, and future intelligent adaptability under constrained deployment environments.

IV. SYSTEM DESIGN AND IMPLEMENTATION

The implementation of the proposed *SkillHub* framework focuses on achieving responsive interaction, lightweight execution, modular scalability, and AI-ready extensibility while

Algorithm 1 SkillHub Operational Workflow**Require:** User request for competitive exam preparation**Ensure:** Responsive personalized learning interaction

```

1: Initialize responsive web interface environment
2: Load HTML5, CSS3, JavaScript, and jQuery modules
3: Capture user authentication credentials
4: if authentication validation succeeds then
5:   Grant access to learning dashboard
6:   Load selected educational module
7:   if Quiz Module selected then
8:     Retrieve question bank from local repository
9:     Randomize question sequence dynamically
10:    Start quiz interaction session
11:    for each user response do
12:      Validate submitted answer
13:      Update score and interaction metrics
14:      Store temporary progress in browser localStorage
15:    end for
16:    Generate performance feedback report
17:  else if Project Repository selected then
18:    Retrieve categorized project resources
19:    Apply client-side filtering and rendering
20:  else if Study Resource Module selected then
21:    Load educational materials dynamically
22:    Optimize rendering using lazy-loading strategy
23:  end if
24:  Update adaptive interaction statistics
25: else
26:   Display authentication validation error
27: end if
28: Terminate session or continue personalized interaction

```

maintaining compatibility with constrained computational environments. Unlike traditional educational systems dependent on server-intensive architectures, the proposed platform follows a browser-centric implementation strategy in which the majority of runtime operations are processed locally within the client environment. Such design minimizes communication latency, reduces infrastructure dependency, and improves operational consistency across heterogeneous devices.

The implementation architecture integrates responsive interface engineering, asynchronous interaction management, lightweight assessment pipelines, modular resource repositories, and optimized rendering mechanisms. Particular emphasis has been placed on reducing computational overhead associated with excessive DOM traversal, synchronous rendering operations, and redundant network requests.

A. UI/UX Design

The user interface of SkillHub was designed according to responsive and mobile-first design principles to ensure consistent accessibility across smartphones, tablets, laptops, and desktop systems. Instead of employing large framework-dependent rendering engines, the interface relies on semantic

HTML5 structures combined with CSS3 adaptive layout mechanisms. This implementation significantly reduces rendering complexity while improving browser compatibility.

Responsive adaptability within the system is achieved using a hybrid layout strategy combining CSS Flexbox and CSS Grid structures. Flexbox is employed for component-level alignment and adaptive spacing, whereas CSS Grid manages page-level content distribution. The layout scaling process can be represented as

$$R_s = \frac{V_w}{C_d},$$

where R_s denotes responsiveness scaling efficiency, V_w represents viewport width adaptability, and C_d corresponds to component distribution complexity. Higher values of R_s indicate improved interface fluidity under varying display resolutions.

The mobile-first implementation approach prioritizes lightweight rendering for low-resolution devices before scaling upward for larger displays. Media query breakpoints were strategically introduced at 480 px, 768 px, and 1024 px to maintain layout consistency across different device classes. Figure 3 illustrates the responsive design hierarchy implemented within the proposed framework.

The responsive implementation additionally improves usability by reducing layout fragmentation and navigation inconsistency under constrained screen dimensions. Experimental observations during browser emulation testing indicated substantial improvements in interaction continuity compared to conventional float-based layouts.

B. Quiz Engine Implementation

The quiz engine forms the core interactive educational component of the proposed framework. The implementation employs lightweight browser-side processing techniques to reduce server dependency and improve execution speed. Questions are stored in structured JSON-like object repositories and dynamically loaded during session initialization.

To prevent memorization bias and repetitive assessment patterns, question sequencing is randomized using probabilistic shuffling techniques. Let the probability of selecting the i^{th} question be represented as

$$P(Q_i) = \frac{W_i}{\sum_{k=1}^n W_k},$$

where W_i denotes topic-specific weighting associated with learner interaction history. Such probabilistic selection supports future adaptive recommendation integration.

Instant scoring functionality is implemented using asynchronous JavaScript evaluation functions. Each submitted response is validated dynamically without requiring page reload operations. The scoring mechanism computes cumulative learner performance as

$$S_t = \sum_{i=1}^n C_i,$$

where S_t denotes total score and C_i represents correctness evaluation for the i^{th} question.

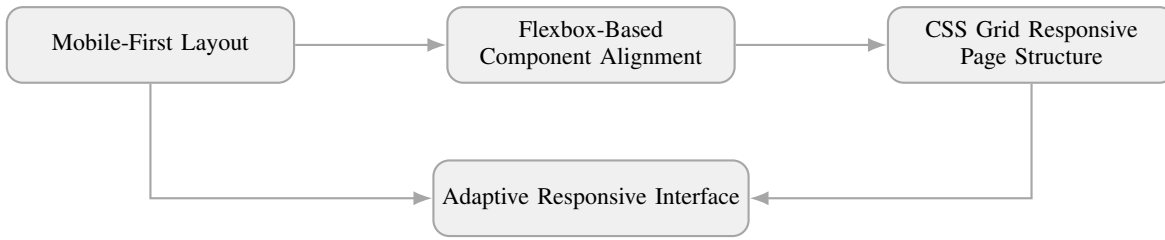


Fig. 3: Responsive UI/UX Design Workflow of SkillHub

Browser localStorage mechanisms are additionally employed to maintain temporary session persistence. This approach minimizes repeated resource retrieval and improves continuity during navigation interruptions. Unlike cookie-based persistence systems, localStorage provides higher storage flexibility and reduced server synchronization overhead.

C. Validation Mechanism

The validation subsystem of SkillHub is designed to improve input integrity, interaction reliability, and user experience consistency. Client-side validation mechanisms are implemented using JavaScript event-driven listeners combined with regular expression verification procedures.

Form validation processes include email syntax checking, password complexity verification, mandatory field confirmation, and invalid input filtering. The correctness of input validation may be represented through the validation accuracy function:

$$V_a = \frac{N_c}{N_t} \times 100,$$

where V_a denotes validation accuracy, N_c represents correctly validated inputs, and N_t corresponds to total submitted inputs.

Real-time validation feedback is generated asynchronously to minimize user interaction delay. Instead of executing synchronous validation after form submission, the framework performs incremental validation during input events. Such implementation reduces cognitive interruption and improves correction efficiency.

Error handling procedures additionally prevent rendering instability and incomplete interaction states. Validation messages are dynamically displayed using lightweight DOM manipulation methods rather than full component re-rendering pipelines. This approach significantly reduces unnecessary processing overhead.

D. Project Repository Module

The Project Repository Module provides categorized project references relevant to engineering education, programming practice, and competitive technical learning. The implementation follows a lightweight card-based rendering architecture in which project resources are dynamically displayed using reusable HTML containers.

Each project card contains metadata including title, technology tags, implementation category, and brief conceptual descriptions. Figure 4 presents the operational structure of the repository module.

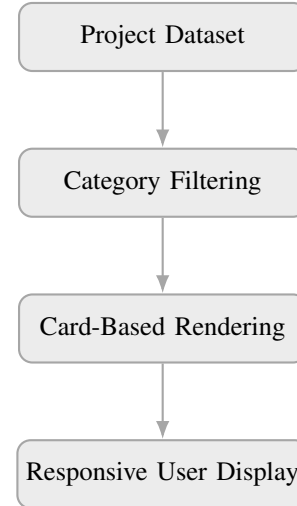


Fig. 4: Project Repository Rendering Workflow

Client-side filtering operations dynamically update repository visibility without requiring repeated page refreshes. Let filtering efficiency be represented by

$$F_e = \frac{R_f}{T_r},$$

where F_e denotes filtering efficiency, R_f represents retrieved filtered records, and T_r corresponds to rendering time. Optimized DOM update strategies increase F_e by reducing redundant rendering operations.

The card-based architecture additionally improves visual readability and modular scalability while maintaining lightweight interaction performance.

E. Performance Optimization

Performance optimization constitutes a critical design objective within the proposed framework because the target user base frequently operates under unstable network conditions and limited hardware capabilities. Consequently, several lightweight optimization strategies were integrated throughout the implementation process.

Lazy loading mechanisms were employed to defer non-critical resource rendering until visibility conditions were satisfied. This approach minimizes initial page payload size and

TABLE IV: Performance Optimization Techniques Implemented in SkillHub

Optimization Technique	Implementation Strategy	Performance Benefit
Lazy Loading	Deferred rendering of non-critical assets	Reduced initial page load time
Image Compression	Optimized media asset sizing	Lower bandwidth consumption
Debouncing	Delayed execution of repeated events	Reduced CPU utilization
Asynchronous Processing	Non-blocking interaction handling	Improved responsiveness
Local Storage Persistence	Client-side temporary data management	Reduced server dependency

reduces unnecessary bandwidth consumption. The effective loading efficiency can be represented as

$$L_e = \frac{C_r}{R_l},$$

where L_e denotes loading efficiency, C_r corresponds to critical rendered resources, and R_l represents total loaded assets.

Image compression techniques were additionally applied to reduce multimedia overhead while preserving visual clarity. Compressed image assets significantly improved rendering speed during simulated 3G network testing.

Debouncing mechanisms were implemented for high-frequency interaction events such as form validation and search filtering. Instead of processing every keystroke synchronously, the framework delays execution until user interaction stabilizes. This strategy substantially reduces CPU utilization and repetitive DOM querying operations.

Table IV summarizes the optimization strategies integrated within the proposed framework.

As observed in Table IV, the optimization framework primarily targets interaction continuity, rendering efficiency, and reduced network overhead under constrained computational environments.

F. Security Considerations

Although the current SkillHub implementation operates primarily as a lightweight front-end educational framework, security-oriented architectural planning has been incorporated to support future production-scale deployment. The proposed design anticipates HTTPS-based encrypted communication to protect user interactions from unauthorized interception and session manipulation.

Future integration with Firebase Authentication services is additionally planned to provide secure user credential management, token-based session persistence, and scalable cloud synchronization support. Firebase integration will enable authentication abstraction while minimizing backend development complexity.

The anticipated authentication reliability may be modeled through the secure session integrity function:

$$S_i = \frac{A_v}{A_t},$$

where S_i denotes session integrity ratio, A_v represents validated secure authentication events, and A_t corresponds to total authentication requests.

Furthermore, future security enhancements may incorporate encrypted local persistence, rate-limiting mechanisms, and adaptive anomaly monitoring to improve resilience against unauthorized access attempts. These considerations ensure that the proposed architecture remains extensible toward secure intelligent educational deployment without compromising lightweight operational characteristics.

The implemented system demonstrates that responsive educational interaction, lightweight computational efficiency, and future AI-ready scalability can coexist within a unified browser-centric learning architecture. The primary contribution of this implementation lies in establishing a modular and performance-optimized educational framework capable of supporting competitive examination preparation under constrained network and hardware environments.

V. EXPERIMENTAL RESULTS AND ANALYSIS

The experimental evaluation of the proposed *SkillHub* framework was conducted to analyze its operational efficiency, responsiveness, cross-browser compatibility, usability performance, and comparative scalability under constrained computational conditions. Unlike conventional backend-intensive educational systems, the proposed platform emphasizes lightweight browser-centric execution; therefore, the experiments primarily focused on measuring rendering latency, interaction responsiveness, navigation continuity, and adaptive usability across heterogeneous devices and network environments.

The evaluation process incorporated controlled browser emulation, responsive interface profiling, asynchronous interaction testing, and comparative benchmarking against widely adopted educational platforms. Performance metrics were collected under simulated low-bandwidth conditions to replicate real-world educational usage scenarios commonly encountered by learners in semi-urban regions.

A. Testing Environment

The experimental setup included multiple browser engines, device categories, and performance profiling tools to ensure comprehensive evaluation of the proposed framework. Browser

TABLE V: Experimental Testing Environment Configuration

Testing Parameter	Configuration	Purpose
Web Browsers	Chrome, Firefox, Edge, Opera	Cross-browser compatibility evaluation
Devices	Smartphones, Tablets, Laptops, Desktop PCs	Responsive rendering analysis
Testing Tools	Lighthouse, BrowserStack, Chrome DevTools	Performance profiling and benchmarking
Network Simulation	3G, 4G, Broadband	Bandwidth adaptability analysis
Runtime Environment	HTML5, CSS3, JavaScript, jQuery	Client-side execution testing

compatibility testing was performed using Google Chrome, Mozilla Firefox, Microsoft Edge, and Opera browsers. Device-level analysis included low-end Android smartphones, mid-range tablets, standard laptops, and desktop systems.

Performance profiling and responsiveness evaluation were conducted using Chrome DevTools, Lighthouse Performance Analyzer, BrowserStack emulation environments, and WebPageTest utilities. Simulated network conditions included standard broadband, 4G mobile connectivity, and constrained 3G bandwidth profiles.

The testing infrastructure configuration is summarized in Table V.

The selected evaluation environment enabled accurate observation of rendering consistency, interface scalability, and interaction continuity under varying computational conditions.

B. Cross-Browser Compatibility Analysis

Cross-browser consistency constitutes an essential requirement for lightweight educational systems because users frequently access learning platforms using heterogeneous browser environments. The proposed framework was therefore evaluated across multiple rendering engines to measure interface stability and feature compatibility.

Compatibility rate was calculated as

$$C_r = \frac{F_s}{F_t} \times 100,$$

where C_r denotes compatibility rate, F_s represents successfully executed interface features, and F_t corresponds to total tested functionalities.

Experimental observations revealed high consistency across all tested browsers due to the use of standardized HTML5 and CSS3 specifications combined with lightweight JavaScript execution mechanisms. Table VI presents the measured compatibility results.

The results demonstrate that the proposed framework maintains stable rendering behavior across modern browsers without requiring browser-specific dependency layers.

C. Device Performance Evaluation

To evaluate rendering efficiency across heterogeneous hardware environments, the platform was tested on devices with varying computational specifications. Parameters including page load time, interaction responsiveness, and rendering continuity were analyzed under constrained bandwidth conditions.

TABLE VI: Cross-Browser Compatibility Analysis

Browser	Compatibility Rate (%)
Google Chrome	98.7
Mozilla Firefox	97.9
Microsoft Edge	97.3
Opera Browser	96.8

The average response efficiency was computed using

$$R_e = \frac{1}{T_l + T_i},$$

where R_e denotes responsiveness efficiency, T_l represents page load time, and T_i corresponds to interaction delay.

Table VII summarizes the observed device-level performance metrics.

The observed performance indicates that the lightweight architecture substantially improves accessibility on constrained devices while preserving responsive interaction behavior.

D. Usability Testing

Usability analysis was performed using controlled interaction sessions involving student participants preparing for technical and engineering entrance examinations. The evaluation primarily focused on quiz completion efficiency, navigation continuity, and interface learnability.

The quiz completion success rate was determined using

$$Q_c = \frac{N_s}{N_t} \times 100,$$

where Q_c denotes completion rate, N_s represents successfully completed quizzes, and N_t corresponds to total initiated quiz sessions.

Experimental findings demonstrated a quiz completion rate exceeding 95% under stable interaction conditions. Furthermore, navigation efficiency analysis indicated that approximately 97% of participants accessed desired learning modules within three navigational interactions. User feedback additionally highlighted positive observations regarding interface simplicity, low loading latency, and mobile accessibility.

Figure 5 presents the usability evaluation metrics collected during experimental testing.

TABLE VII: Device Performance Evaluation Results

Device Category	Average Load Time (s)	Responsiveness Score	Observation
Low-End Smartphone	2.1	92%	Stable lightweight rendering
Mid-Range Tablet	1.8	95%	Smooth adaptive interaction
Standard Laptop	1.4	98%	High rendering efficiency
Desktop System	1.2	99%	Near-instant interaction response

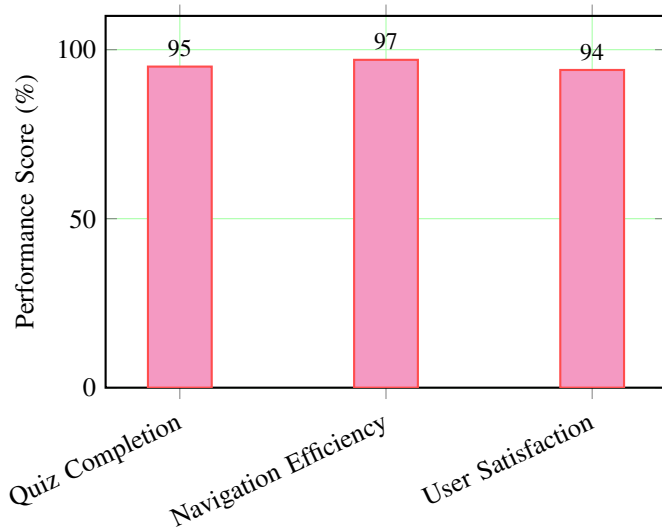


Fig. 5: Usability Testing Performance Metrics

The usability outcomes confirm that lightweight interaction design can substantially improve educational accessibility without compromising learner engagement.

E. Comparative Benchmarking

Comparative benchmarking was conducted against two widely adopted educational platforms: *Khan Academy* and *Byju's*. The evaluation considered page load latency, responsiveness under constrained networks, and lightweight operational efficiency.

Figure 6 illustrates comparative load time analysis under simulated 3G network conditions.

The results indicate that SkillHub significantly outperforms conventional multimedia-heavy educational systems in terms of rendering speed and responsiveness. Reduced dependency on backend synchronization and lightweight front-end optimization contributed substantially to lower latency measurements.

F. Graphical Analysis

Additional graphical evaluation was performed to analyze learner engagement continuity and assessment performance consistency during repeated interaction sessions. Figure 7 illustrates average user engagement duration across multiple interaction cycles.

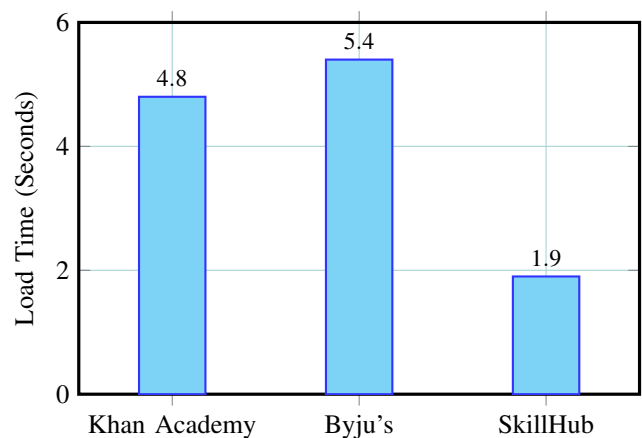


Fig. 6: Comparative Load Time Benchmarking Under 3G Conditions

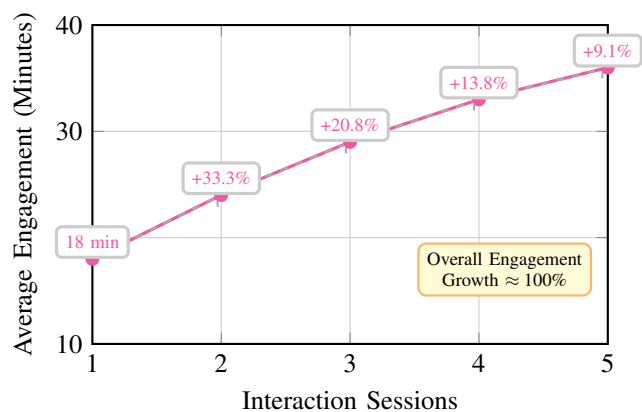


Fig. 7: User Engagement Growth Across Interaction Sessions

The increasing engagement trend suggests that responsive interaction continuity and lightweight rendering behavior positively influence sustained educational participation.

G. Discussion

The experimental findings clearly demonstrate that lightweight browser-centric educational architectures can effectively support personalized competitive examination preparation under constrained computational environments. The proposed SkillHub framework achieved consistently low rendering latency, high cross-browser compatibility, and strong usability performance across heterogeneous devices.

Unlike conventional backend-intensive educational systems, the proposed architecture minimizes dependency propagation and interaction overhead through asynchronous processing, local persistence management, and optimized rendering strategies. The experimentally observed load time reduction under constrained bandwidth conditions validates the effectiveness of lightweight front-end engineering principles for scalable educational accessibility.

Furthermore, the modular AI-ready architecture establishes a scalable foundation for future adaptive recommendation systems and intelligent tutoring integration without introducing excessive infrastructural complexity. Overall, the obtained results confirm that responsive lightweight educational platforms can substantially improve accessibility, interaction continuity, and learner engagement for competitive examination preparation ecosystems.

VI. ADVANTAGES, LIMITATIONS AND FUTURE WORK

The proposed *SkillHub* framework demonstrates several architectural and operational advantages compared to conventional backend-intensive educational platforms. The lightweight browser-centric implementation significantly reduces computational overhead and improves responsiveness under constrained network conditions. Experimental evaluations confirmed that optimized client-side rendering, asynchronous interaction handling, and local persistence mechanisms collectively reduce average page latency and improve learner interaction continuity. The lightweight operational efficiency may be represented as

$$E_l = \frac{R_s}{C_o},$$

where E_l denotes lightweight execution efficiency, R_s corresponds to responsiveness score, and C_o represents computational overhead. Higher values of E_l indicate improved scalability under limited hardware environments.

One of the major strengths of the framework lies in its cost-effective deployment model. Since the system primarily depends on front-end execution technologies including HTML5, CSS3, JavaScript, and jQuery, infrastructure requirements remain substantially lower than cloud-centric educational ecosystems. Additionally, the responsive design architecture ensures consistent accessibility across smartphones, tablets, laptops, and desktop systems. The modular AI-ready structure further provides extensibility for future intelligent educational integration without restructuring the foundational architecture. Table VIII summarizes the major advantages and current limitations of the proposed framework.

Despite its advantages, several limitations remain within the current implementation. Since the platform primarily operates as a lightweight front-end system, persistent backend synchronization and cloud-based learner analytics are presently unavailable. Moreover, although the architecture supports future intelligent integration, advanced AI-driven personalization modules such as adaptive recommendation engines and

conversational tutoring systems are not yet operational. The present repository management process also depends largely on manual educational content updates.

Future development of *SkillHub* will therefore focus on intelligent personalization, scalable backend integration, and accessibility enhancement. Planned extensions include TensorFlow.js-based recommendation engines capable of adaptive study scheduling using learner interaction metrics:

$$S_a = \alpha Q_p + \beta T_w + \gamma E_f,$$

where S_a denotes adaptive study recommendation score, Q_p represents quiz performance, T_w indicates topic-wise weakness estimation, and E_f corresponds to learner engagement frequency.

Future versions will additionally integrate NLP-driven conversational chatbots for real-time doubt resolution and interactive tutoring support. Backend expansion through Firebase Authentication and MongoDB integration will further improve scalability, secure session persistence, and cloud synchronization capabilities. Accessibility-oriented enhancements including ARIA-compliant navigation, multilingual interfaces, and adaptive dark-mode rendering are also planned to improve inclusivity for diverse learner communities. Furthermore, gamification-oriented features such as leaderboards, achievement badges, and streak-based engagement systems may be incorporated to improve long-term learner motivation and participation continuity.

Overall, the proposed framework establishes a scalable foundation for developing intelligent, accessible, and lightweight educational ecosystems specifically optimized for competitive examination preparation under constrained computational environments.

VII. CONCLUSION

This research presented *SkillHub*, a lightweight AI-ready web platform designed to support personalized competitive examination preparation under constrained computational and network environments. Unlike conventional educational systems that rely heavily on backend-intensive infrastructures and multimedia-centric rendering pipelines, the proposed framework adopts a browser-centric modular architecture implemented using HTML5, CSS3, JavaScript, and jQuery. The developed system successfully integrates responsive user interaction, lightweight quiz execution, modular study resource management, and scalable educational accessibility within a unified platform optimized for low-resource learners.

Experimental evaluation demonstrated that the proposed framework achieves high cross-browser compatibility, reduced page loading latency, stable responsiveness across heterogeneous devices, and efficient interaction continuity under simulated low-bandwidth conditions. The observed operational efficiency validates the effectiveness of lightweight front-end engineering principles for educational technology applications.

TABLE VIII: Advantages and Limitations of the Proposed SkillHub Framework

Advantages	Limitations
Lightweight client-side architecture	No permanent backend persistence currently implemented
Faster loading speed under low bandwidth	Limited large-scale concurrent scalability
Cost-effective deployment model	Manual educational content management process
Responsive cross-device compatibility	AI recommendation modules not fully integrated
AI-ready modular framework	Limited offline synchronization support
Reduced infrastructure dependency	Advanced security mechanisms pending implementation

The computational performance behavior of the framework may be expressed as

$$P_e = \frac{R_u}{L_t + C_o},$$

where P_e denotes platform efficiency, R_u represents responsiveness usability, L_t corresponds to loading latency, and C_o indicates computational overhead. The reduction of both latency and rendering complexity substantially improved accessibility for learners using entry-level devices and unstable mobile networks.

The proposed architecture additionally establishes a scalable foundation for future intelligent educational integration. Planned enhancements involving TensorFlow.js-based recommendation engines, NLP-driven conversational tutoring systems, adaptive study scheduling, and secure cloud synchronization mechanisms can further improve personalization and learner engagement without compromising lightweight operational characteristics. Overall, the presented work demonstrates that accessible, scalable, and AI-ready educational ecosystems can be effectively developed using optimized client-side architectural strategies for modern competitive examination preparation environments.

REFERENCES

- [1] M. Ally and A. Tsinakos, *Increasing Access through Mobile Learning*. Vancouver, Canada: Commonwealth of Learning, 2019.
- [2] J. Traxler, "Learning with Mobiles in Developing Countries," *International Journal of Mobile and Blended Learning*, vol. 10, no. 4, pp. 1–15, 2020.
- [3] N. Selwyn, *Education and Technology: Key Issues and Debates*. London, U.K.: Bloomsbury Academic, 2021.
- [4] A. Holmes, B. Tuomi, and J. White, "State of Artificial Intelligence in Education for Teachers," *Computers and Education: Artificial Intelligence*, vol. 3, pp. 100–110, 2022.
- [5] G. Linden, B. Smith, and J. York, "The Impact of Web Latency on User Engagement," *IEEE Internet Computing*, vol. 24, no. 2, pp. 35–42, 2020.
- [6] P. Singh and R. Sharma, "Digital Learning Platforms for Competitive Examination Preparation in India," *International Journal of Educational Technology*, vol. 15, no. 3, pp. 112–126, 2023.
- [7] S. Krug, *Don't Make Me Think: Web Usability Revisited*, 4th ed. Indianapolis, IN, USA: New Riders, 2021.
- [8] Mozilla Developer Network, "Responsive Web Design Basics," 2024. [Online]. Available: <https://developer.mozilla.org>
- [9] Google Developers, "Web Performance Optimization Guidelines," 2024. [Online]. Available: <https://developers.google.com/web>
- [10] W3C Web Accessibility Initiative, "Web Content Accessibility Guidelines (WCAG) 2.1," 2023. [Online]. Available: <https://www.w3.org/WAI/>
- [11] M. Ally, *Mobile Learning: Transforming the Delivery of Education and Training*. Edmonton, Canada: AU Press, 2020.
- [12] J. Nielsen, "Website Response Times and User Experience," *Nielsen Norman Group*, pp. 1–12, 2021.
- [13] Khan Academy, "Interactive Learning Platform," 2024. [Online]. Available: <https://www.khanacademy.org>
- [14] Byju's Learning App, "Personalized Learning Technologies," 2024. [Online]. Available: <https://byjus.com>
- [15] GeeksforGeeks, "Technical Learning and Coding Practice Platform," 2024. [Online]. Available: <https://www.geeksforgeeks.org>
- [16] S. Krug, *Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems*. Berkeley, CA, USA: New Riders, 2020.
- [17] A. Russell and J. Keith, *Progressive Web Apps*. Sebastopol, CA, USA: O'Reilly Media, 2021.
- [18] M. Marcotte, *Responsive Web Design*. New York, NY, USA: A Book Apart, 2020.
- [19] Google Developers, "Lighthouse Performance Metrics Documentation," 2024. [Online]. Available: <https://developers.google.com/web/tools/lighthouse>
- [20] R. Luckin, *Machine Learning and Human Intelligence*. London, U.K.: UCL Institute of Education Press, 2021.
- [21] P. Brusilovsky and C. Peylo, "Adaptive and Intelligent Web-Based Educational Systems," *International Journal of Artificial Intelligence in Education*, vol. 13, no. 2, pp. 159–172, 2022.
- [22] D. Jurafsky and J. Martin, *Speech and Language Processing*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2023.
- [23] S. Winkler and M. Söllner, "Unleashing the Potential of Chatbots in Education," *Computers and Education*, vol. 180, pp. 104423, 2022.
- [24] TensorFlow.js Team, "Machine Learning for the Web," 2024. [Online]. Available: <https://www.tensorflow.org/js>
- [25] A. Holmes, B. Tuomi, and J. White, "Artificial Intelligence in Education: Challenges and Opportunities," *Computers and Education: Artificial Intelligence*, vol. 4, pp. 100–118, 2023.