



(RESEARCH ARTICLE)



Attention Guardian AI

Shashank Tiwari, Sai Karthik Sriram, Shiva Sai Yelagala and Yashwanth Gujja *

Department of CSE (AI and ML) of ACE Engineering College, Hyderabad, India.

World Journal of Advanced Research and Reviews, 2026, 30(01), 213-219

Publication history: Received on 22 February 2026; revised on 1 April 2026; accepted on 4 April 2026

Article DOI: <https://doi.org/10.30574/wjarr.2026.30.1.0797>

Abstract

Online learning platforms are widely used for education, but frequent advertisements distract users and reduce learning efficiency. This paper presents Attention Guardian AI, a Chrome extension that detects advertisements and replaces them with study tips and interactive quizzes. The system uses DOM manipulation and MutationObserver for real-time ad detection and analyzes webpage content to provide relevant educational material. AI integration enables personalized content generation, while offline support ensures continuous functionality. Additionally, user activity is tracked and visualized through a dashboard. The proposed system enhances focus, improves productivity, and promotes active learning.

Keywords: Attention Guardian AI; Chrome Extension; Artificial Intelligence; Ad Replacement; Online Learning; DOM Manipulation; Personalized Learning; Learning Analytics

1. Introduction

In recent years, the rapid growth of digital platforms has increased students' online learning time. While websites like YouTube, W3Schools, and GeeksforGeeks provide valuable content, frequent advertisements distract users and reduce productivity. To address this issue, the project Attention Guardian AI aims to convert distractions into learning opportunities. It is a Chrome extension that detects ads and replaces them with AI-generated study tips and quizzes.

The system uses DOM manipulation and MutationObserver to identify and replace ads in real time. It analyzes webpage content to provide relevant tips based on the topic being studied. When connected to an AI server, it generates personalized tips and quizzes; otherwise, it shows preloaded content in offline mode.

Additionally, the system tracks user activity such as topics learned and quizzes attempted. This data is displayed in a React-based dashboard to monitor progress. Overall, the project enhances focus, improves learning efficiency, and promotes productive online study.

2. Literature Survey

2.1. S. Ikram & M. A. Kaafar (2017) – Ad Blockers and Their Impact on User Experience

Ikram and Kaafar (2017) studied the impact of ad blockers on user experience by analyzing how intrusive advertisements affect browsing behavior. Their work highlights that advertisements significantly reduce user attention and efficiency during online activities. Ad blockers help in removing these distractions and improve browsing experience. However, they only eliminate ads and do not provide any additional benefit to the user. This limitation creates an opportunity for systems that can replace ads with useful and meaningful content, as proposed in this project.

* Corresponding author: Yashwanth Gujja

2.1.1. Methodologies and Algorithms

The methodology focuses on detecting advertisement elements using filtering techniques and blocking them at the browser level. Algorithms identify ad-related scripts and remove them before rendering the webpage. These approaches improve usability but lack value-added content replacement. In the proposed system, this concept is extended by not only removing ads but replacing them with study tips and quizzes.

2.2. R. S. Baker & K. Yacef (2009) – Artificial Intelligence in Personalized Learning Systems

Baker and Yacef (2009) explored the use of artificial intelligence in creating personalized learning environments. Their research shows how AI systems analyze user behavior, preferences, and interactions to provide customized learning content such as recommendations, tips, and assessments. This approach improves engagement and knowledge retention by adapting content to individual needs. The proposed project applies this concept by generating personalized study tips and quizzes based on the user's browsing context.

2.2.1. Methodologies and Algorithms

The methodology involves collecting user data and applying machine learning algorithms to analyze learning patterns. AI models generate personalized outputs based on user interactions and content relevance. These systems rely on data processing and adaptive algorithms. In the proposed system, AI APIs are used to generate dynamic study content, with fallback mechanisms for offline usage.

2.3. D. Flanagan (2011) – Dynamic Web Content Manipulation using DOM and Event Observers

Flanagan (2011) focused on techniques for dynamically modifying web content using the Document Object Model (DOM) and event listeners such as MutationObserver. The study explains how web elements can be detected, modified, or replaced in real time without disrupting the functionality of the webpage. This concept is directly applied in the proposed project to detect advertisement elements and replace them with educational content.

2.3.1. Methodologies and Algorithms

The methodology includes using DOM traversal and event-driven programming to monitor changes in web pages. MutationObserver is used to track dynamically loaded elements. Algorithms identify specific HTML structures and modify them accordingly. In the proposed system, these techniques are used to detect ads and inject study tips and quizzes in real time.

2.4. J. Roediger & H. Karpicke (2006) – Effectiveness of Quiz-Based Learning Systems

Roediger and Karpicke (2006) studied the role of quizzes in improving learning outcomes. Their research shows that active recall through quizzes enhances memory retention and understanding compared to passive reading. Frequent testing helps learners engage more deeply with content and improves long-term knowledge retention. This concept is incorporated into the proposed system through interactive quiz overlays.

2.4.1. Methodologies and Algorithms

The methodology involves designing structured quizzes that test user knowledge and reinforce learning. Algorithms evaluate responses and provide feedback. These systems focus on active learning strategies. In the proposed project, quizzes are displayed during ad interruptions to convert passive time into active learning.

2.5. A. B. Brush, B. Lee & S. Mahajan (2015) – Browser Extensions for Enhancing Productivity

Brush, Lee, and Mahajan (2015) discussed how browser extensions can improve productivity by modifying user interaction with web content. Their research highlights how extensions can block distractions, provide reminders, and enhance workflow efficiency. However, such systems are limited by browser permissions and compatibility issues. The proposed project builds on this idea by creating an extension that not only blocks distractions but also enhances learning.

2.5.1. Methodologies and Algorithms

The methodology includes using browser APIs and content scripts to interact with web pages. Event-driven mechanisms are used to trigger actions based on user activity. Algorithms manage extension behavior and content injection. In the proposed system, similar techniques are used to detect ads and replace them with useful learning content.

Table 1 Comparison Table of Literature Survey

S. No	Author(s)	Title	Methodology Used	Findings from the Reference Paper
1	S. Ikram & M. A. Kaafar (2017)	Ad Blockers and Their Impact on User Experience	Used filtering techniques to detect and block advertisement elements at browser level	Improved browsing experience by removing distractions but did not provide additional value to users
2	R. S. Baker & K. Yacef (2009)	Artificial Intelligence in Personalized Learning Systems	Applied AI algorithms to analyze user behavior and generate personalized content	Enhanced engagement and learning efficiency but requires reliable data and computation
3	D. Flanagan (2011)	Dynamic Web Content Manipulation using DOM and Event Observers	Used DOM manipulation and MutationObserver to detect and modify webpage elements in real time	Enabled dynamic content updates but implementation can be complex across websites
4	J. Roediger & H. Karpicke (2006)	Effectiveness of Quiz-Based Learning Systems	Designed quiz-based learning strategies to improve memory through active recall	Improved knowledge retention and engagement but requires well- structured questions
5	A. B. Brush, B. Lee & S. Mahajan (2015)	Browser Extensions for Enhancing Productivity	Used browser APIs and content scripts to modify user interaction with web content	Increased productivity but limited by browser permissions and compatibility

3. System Architecture

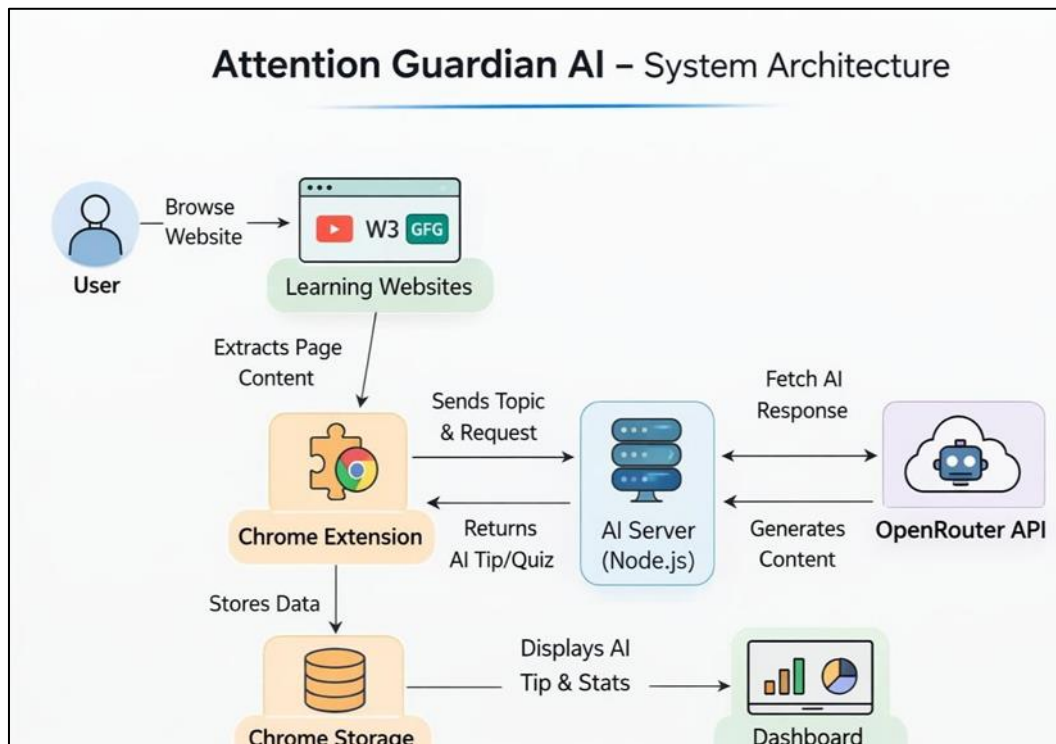


Figure 1 System Architecture

The system architecture of the Attention Guardian AI is designed based on three main layers: User Layer, Application Layer, and Processing Layer, which together enable intelligent content filtering and learning enhancement while browsing educational websites.

The User Layer represents the interaction of the user with the system. In this project, the user browses learning platforms such as YouTube, W3Schools, and GeeksforGeeks using a Chrome browser. The user does not directly control the system but interacts naturally by viewing content. The Chrome Extension works in the background and enhances the browsing experience by replacing distracting ads with study tips and quizzes. This layer acts as the entry point of the system.

The Application Layer acts as the core logic of the system. It is implemented using JavaScript within the Chrome Extension and Node.js on the server side. The extension extracts page content, detects the topic and concept using DOM analysis, and sends requests to the AI Server. The AI Server processes these requests and communicates with the OpenRouter API to generate relevant study tips and quizzes. The extension then displays these tips or quizzes to the user in place of ads. Additionally, this layer manages user interaction data and updates learning statistics.

The Processing Layer is responsible for handling data processing and AI-based content generation. The extracted webpage content is analyzed to identify topics such as HTML, CSS, JavaScript, or Data Structures. The AI Server uses this information to generate meaningful tips and quizzes through the OpenRouter API. In offline mode, predefined quizzes are randomly selected and displayed. The Chrome Storage module stores user activity such as number of ads blocked, quizzes attempted, and topics explored. The React Dashboard retrieves this data and presents it in a visual format.

The system runs continuously using event-driven mechanisms such as DOM selectors and MutationObserver, which detect changes in the webpage dynamically. This allows the extension to identify ads in real time and replace them with educational content instantly. The architecture ensures smooth interaction between user activity, intelligent processing, and real-time feedback, making the system efficient, adaptive, and helpful for focused learning.

4. Proposed Methodology

4.1. Chrome Extension Architecture

The first step in the methodology involves designing the Chrome extension structure using Manifest V3. The extension consists of key components such as content scripts, background scripts, popup interface, and React dashboard.

The content script is responsible for interacting with web pages, while the background script manages events and communication. The popup provides quick user interaction, and the dashboard displays analytics. This architecture forms the foundation of the Attention Guardian AI system.

4.2. Ad Detection and Content Analysis

The system identifies advertisements on platforms such as YouTube, W3Schools, and GeeksforGeeks using DOM manipulation techniques. Specific HTML elements related to ads are detected and targeted.

MutationObserver is used to monitor dynamic changes in the webpage, ensuring ads are detected even when loaded asynchronously. The system also analyzes webpage content such as titles, headings, and keywords to understand the current topic being studied.

4.3. Content Replacement and AI Integration

Once advertisements are detected, they are replaced with educational content such as study tips and quizzes. These tips are generated based on the identified topic to ensure relevance.

When the backend AI server is available, it generates personalized tips and quizzes dynamically. In offline mode, the system displays preloaded quizzes and tips, ensuring uninterrupted functionality. This transforms distractions into productive learning interactions.

4.4. User Tracking and Dashboard Visualization

The system tracks user learning activities such as topics viewed, study tips displayed, and quizzes attempted. This data is stored locally or in the backend database.

A React-based dashboard is used to visualize this information through charts and summaries. This helps users monitor their learning progress, understand their behavior, and stay motivated. The system continuously updates data to provide real-time insights.

5. Algorithmic Flow of the System

The working of the Attention Guardian AI system is explained through the following steps:

- Initialize the Chrome extension with required scripts (content, background, popup, and dashboard).
- Load the webpage and activate the content script.
- Detect advertisement elements using DOM queries.
- Use MutationObserver to track dynamically loaded ads.
- Analyze webpage content to identify the current learning topic.
- Replace detected ads with study tips or quiz overlays.
- If AI server is available:
 - Generate personalized tips and quizzes based on topic.
- Else (offline mode):
 - Display preloaded study tips and quizzes.
- Track user interactions such as tips viewed and quizzes attempted.
- Store user activity data for analysis.
- Display insights on the React dashboard (topics, activity trends, progress).
- Continuously monitor and update the system for real-time functionality.

6. Output Screens

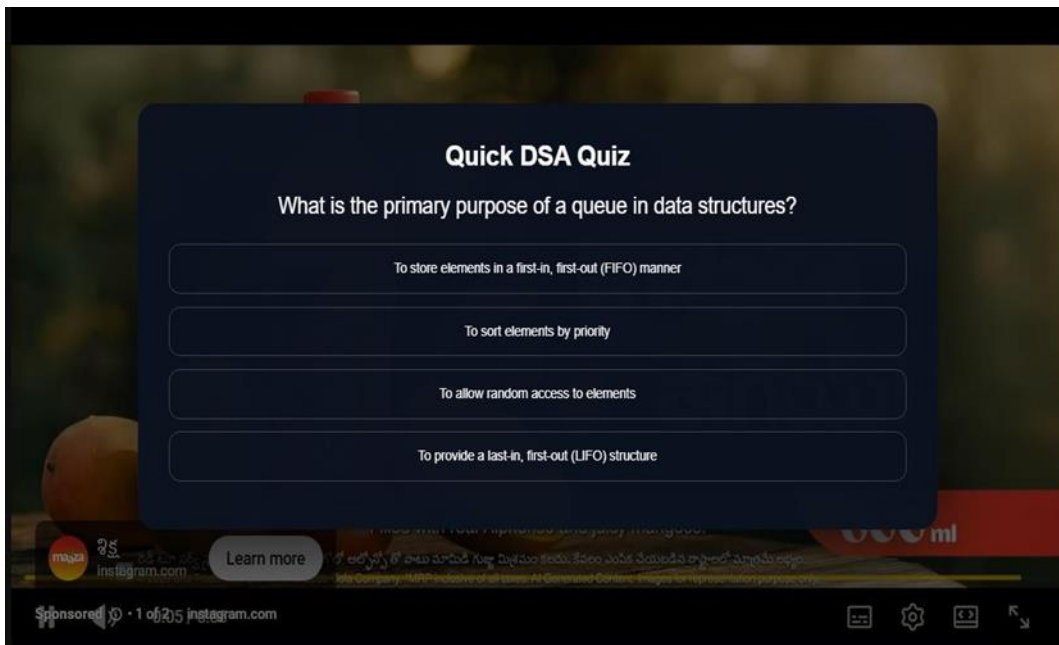


Figure 2 Overall Output of Attention Guardian AI (Ad Replacement with Study Tip)

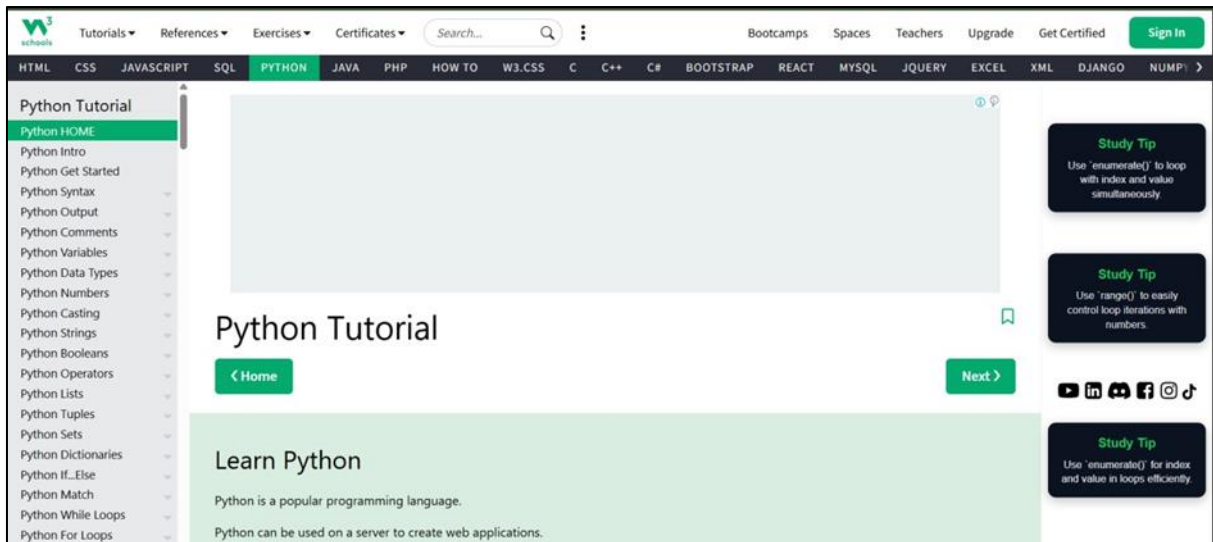


Figure 3 Detection of Webpage Topic and Display of Study Tip

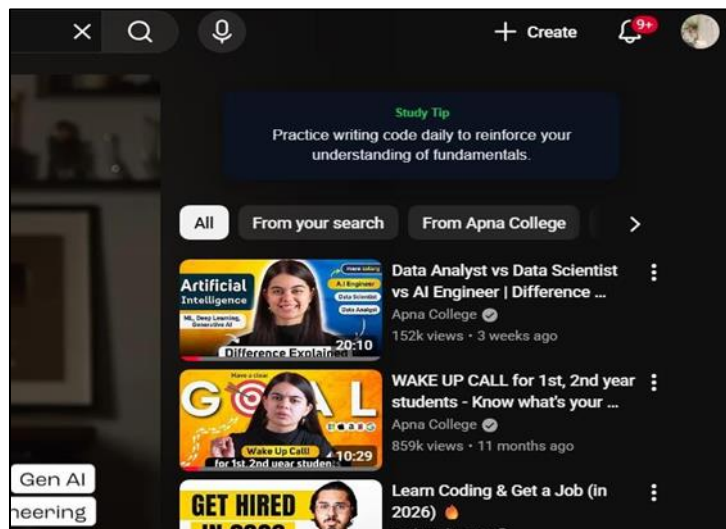


Figure 4 Replacement of side Ad bar with Study Tip

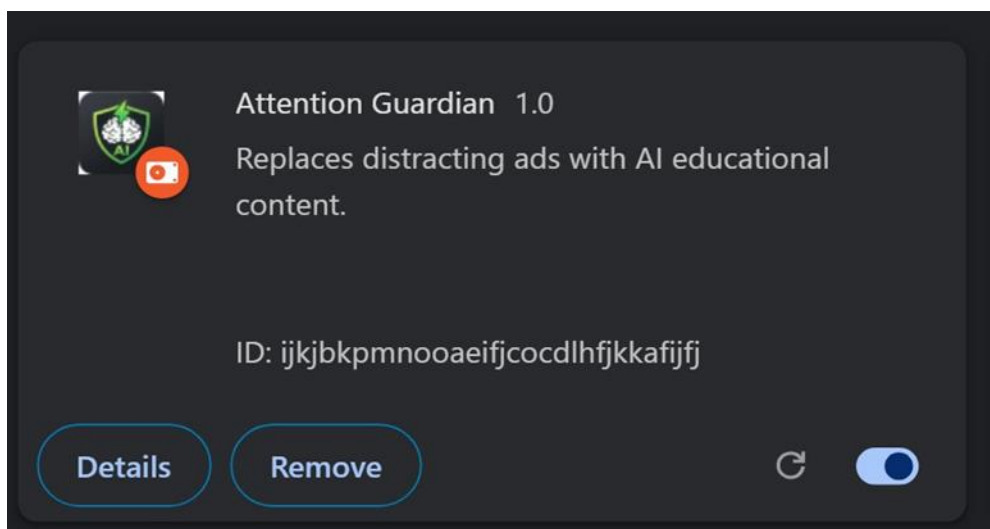


Figure 5 Chrome Extension



Figure 6 Tracking of User Learning Activity using Chrome Storage

7. Conclusion

The Attention Guardian AI system was successfully developed to improve user productivity by converting distractions into learning opportunities. It combines browser automation, AI integration, and real-time processing to enhance the online learning experience. The system detects webpage content and replaces advertisements with relevant study tips. It also introduces interactive quiz overlays during video ads, turning idle time into productive learning.

AI integration enables dynamic generation of personalized tips and quizzes based on browsing context. In offline mode, the system continues to function using preloaded quizzes, ensuring reliability. The project also tracks user activity such as topics explored and quizzes attempted.

This data is visualized through a dashboard, helping users monitor progress and stay motivated. The development process provided experience in Chrome extensions, DOM manipulation, and AI integration. Overall, the system improves focus and learning efficiency. With future enhancements, it can evolve into a powerful AI-based learning assistant.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

- [1] Google Developers, "Chrome Extensions Documentation," Available: <https://developer.chrome.com/docs/extensions/>
- [2] Mozilla Developer Network (MDN), "Document Object Model (DOM)," Available: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model
- [3] Node.js Foundation, "Node.js Documentation," Available: <https://nodejs.org/>
- [4] Express.js, "Fast, unopinionated, minimalist web framework for Node.js," Available: <https://expressjs.com/>
- [5] OpenRouter, "OpenRouter API for AI Models," Available: <https://openrouter.ai/>
- [6] React Documentation, "React – A JavaScript library for building user interfaces," Available: <https://react.dev/>
- [7] W3Schools, "Web Development Tutorials," Available: <https://www.w3schools.com/>