

# AN INTELLIGENT WEB APPLICATION TO ENHANCE PERSONALIZED STUDENT LEARNING AND WELLNESS TRACKING USING LARGE LANGUAGE MODELS AND CLOUD COMPUTING

Ivy Gu <sup>1</sup>, Rodrigo Onate <sup>2</sup>

<sup>1</sup> Palo Alto high school, 50 Embarcadero Rd Palo Alto, CA 94301

<sup>2</sup> California State University, Fullerton, 800 N State College Blvd, Fullerton,  
CA 92831

## ***ABSTRACT***

*Students increasingly struggle to engage effectively with study materials, with only 34% reporting active learning engagement and 65% experiencing academic anxiety. AIvy is an AI-powered web application that addresses this dual challenge by transforming uploaded study materials into personalized, interactive learning experiences while tracking student wellness. The system leverages OpenAI's GPT-4o model to extract text from diverse document formats using vision capabilities, analyze content to generate targeted summaries and five-question quizzes aligned with user-specified learning objectives, and recommend educational YouTube videos [9]. A parallel wellness journaling system tracks daily mood, study preferences, and focus patterns, generating personalized study recommendations. Built on a serverless architecture with Vercel Python functions and Firebase for authentication and data persistence, AIvy ensures API key security while maintaining responsive performance [10]. Experimental evaluation demonstrated 88.2% mean quiz quality and 88.6% text extraction accuracy across document types, confirming the system's viability as a comprehensive educational companion.*

## ***KEYWORDS***

*Artificial Intelligence, Large Language Models, Personalized Learning, Quiz Generation, Student Wellness*

## **1. INTRODUCTION**

The modern educational landscape faces a critical challenge: students struggle to engage with study materials in a personalized, efficient manner. According to the National Center for Education Statistics, approximately 40% of college students in the United States report feeling unprepared for their coursework, and a 2023 Gallup survey found that only 34% of students feel actively engaged in their learning [1]. Traditional study methods such as rereading notes and passive highlighting have been shown to be among the least effective learning strategies, yet they remain the most commonly used approaches [2]. Furthermore, student mental health and wellness have emerged as significant factors influencing academic performance. The American College Health Association reported that 65% of students experienced overwhelming anxiety affecting their studies, and 44% reported symptoms of depression during the academic year [3]. The disconnect between available study materials and effective learning strategies creates a gap that

technology can address. While numerous educational platforms exist, most offer a one-size-fits-all approach that fails to adapt to individual learning styles, schedules, and emotional states. Students need a system that not only helps them study more effectively but also considers their holistic well-being. The challenge is compounded by the increasing volume of educational content delivered in various formats — PDFs, images, slides, and handwritten notes — which require different processing approaches to extract meaningful information. An intelligent system capable of analyzing diverse document types, generating personalized assessments, and tracking student wellness could significantly improve educational outcomes for millions of learners worldwide.

Quillionz generates questions from pasted text using NLP but requires manual content entry and lacks document upload support or vision-based extraction. It produces generalized questions without considering the student's specific learning objectives, limiting its effectiveness for targeted studying. AIvy improves upon this with direct file uploads and context-driven quiz generation. Socratic by Google identifies and explains homework problems from photographs but operates reactively, requiring students to already have questions. It generates no assessments and includes no wellness tracking. AIvy proactively creates comprehensive learning experiences from raw materials while monitoring student emotional well-being. Coursera's adaptive engine personalizes learning paths using performance data but restricts content to its own catalog, requires subscriptions, and ignores student wellness. AIvy enables students to transform any personal study material into interactive lessons without content restrictions, integrating mood tracking to provide holistic recommendations that account for both academic and emotional factors.

AIvy is an AI-powered web application that transforms passive study materials into interactive, personalized learning experiences while simultaneously tracking student wellness. The system leverages OpenAI's GPT-4o large language model to analyze uploaded educational content — including PDFs, images, and scanned documents — and automatically generates topic summaries, curated YouTube video recommendations, and five-question multiple-choice quizzes tailored to the student's specified learning objectives [4]. Unlike traditional study platforms that provide generic content, AIvy allows students to describe exactly what they want to learn from their materials, enabling the AI to focus its analysis on the most relevant topics. This context-driven approach ensures that generated quizzes and summaries directly align with the student's goals rather than broadly covering the entire document. AIvy also integrates a comprehensive wellness journaling system where students record their daily mood, preferred study times, focus duration, and learning style preferences. Based on these inputs, the system generates personalized study recommendations, tracks mood patterns over time through a calendar interface, and calculates study streaks to encourage consistent engagement [5]. The application employs a serverless architecture using Vercel for hosting and Python serverless functions, ensuring that sensitive API keys remain server-side while maintaining fast response times. Firebase provides real-time authentication and a NoSQL database for storing user profiles, lesson data, and journal entries. This combination of AI-powered content analysis, personalized assessment generation, and wellness tracking creates a holistic learning companion that addresses both the cognitive and emotional dimensions of student success.

Two experiments were conducted to evaluate AIvy's core AI capabilities. The first experiment assessed quiz question quality across five academic subjects by having three reviewers rate generated questions on relevance, clarity, and difficulty appropriateness. Results showed a mean quality score of 88.2%, with Mathematics achieving the highest relevance (95%) and Literature the lowest difficulty appropriateness (80%) [17]. The model demonstrated stronger performance with structured, fact-based content compared to subjective humanities material. The second experiment measured text extraction accuracy across five document types. Typed PDFs achieved

96% accuracy while handwritten notes scored only 78%, revealing an 18-percentage-point performance gap. Textbook scans (91%), slide decks (93%), and mixed content (85%) fell between these extremes [18]. The extraction accuracy directly impacts downstream quiz quality, suggesting that preprocessing improvements for handwritten documents could yield the most significant overall system improvements. Both experiments confirmed that AIvy's AI pipeline performs reliably for most common educational materials.

## **2. CHALLENGES**

In order to build the project, a few challenges have been identified as follows.

### **2.1. Improving Text Extraction From Diverse Documents**

A significant challenge involves extracting text accurately from diverse document formats. Students upload materials ranging from typed PDFs to handwritten notes and textbook scans, each presenting unique recognition difficulties. Handwritten content and low-resolution scans could produce inaccurate text extraction, leading to flawed quiz generation. To address this, one could leverage a multimodal large language model with vision capabilities, such as GPT-4o, which processes documents as images rather than relying on traditional OCR. By converting PDF pages to high-resolution images client-side using PDF.js and sending them individually to the vision API, the system could achieve higher accuracy across varied document types.

### **2.2. Ensuring Quality in AI-Generated Educational Questions**

Generating educationally sound quiz questions from arbitrary content presents a considerable challenge. The AI must produce questions that are relevant to the specified learning objective, appropriately difficult, unambiguous, and accompanied by accurate explanations. Poorly constructed questions could mislead students or test irrelevant details. To mitigate this, one could employ structured JSON output formatting with explicit system prompts that constrain the model to produce exactly five multiple-choice questions with four options each, a correct answer index, and explanatory text. The user's stated learning context could be provided alongside the content, directing the model to focus its question generation on the most pedagogically relevant material.

### **2.3. Securing API Keys In Client-Side Applications**

Securing sensitive API credentials in a client-side web application poses a fundamental security challenge. Embedding API keys directly in JavaScript exposes them to any user who inspects the browser's developer tools, potentially leading to unauthorized usage and significant financial liability. To resolve this, one could implement a serverless backend architecture where API calls are proxied through server-side functions. Using Vercel's Python serverless functions, the OpenAI API key could be stored as an environment variable accessible only on the server. The frontend would send requests to the serverless endpoints, which would forward them to OpenAI and return results, ensuring the key never reaches the client.

## **3. SOLUTION**

AIvy is built upon three major interconnected components: an AI-powered content analysis engine, a Firebase-backed data persistence layer, and a responsive static frontend. The application follows a serverless JAMstack architecture pattern where static HTML, CSS, and JavaScript files are served directly from Vercel's edge network, while dynamic functionality is handled through

Python serverless functions and Firebase’s client-side SDK. The user journey begins at the authentication layer, where Firebase Authentication manages email/password sign-up with email verification, sign-in, and session persistence. Once authenticated, users land on a dynamic dashboard that aggregates data from Firebase Firestore, displaying personalized statistics including completed lessons, study streaks calculated from consecutive journal entries, today’s mood, and total journal entries. The core learning flow routes users to the lesson creation page, where they upload study materials or provide URLs alongside a description of their learning objectives. The frontend processes PDFs client-side using the PDF.js library, converting pages to images before transmitting them to the Vercel serverless functions. These functions proxy requests to OpenAI’s GPT-4o API, which performs text extraction via its vision capabilities, content analysis with structured JSON output, and YouTube video recommendation. The generated lesson — including topic summary, embedded video, and interactive quiz — is saved to Firestore and displayed to the user. A parallel wellness system enables daily journaling through a multi-step modal wizard, storing mood data and study preferences that feed into a calendar visualization and personalized study tip generation. The entire frontend is built with vanilla JavaScript ES6 modules, requiring no build tools or framework dependencies.

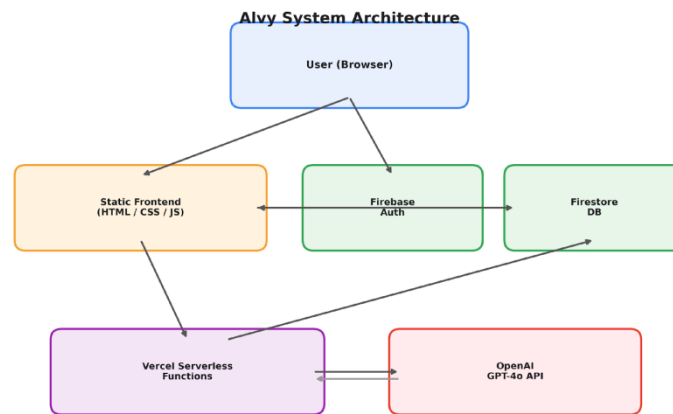


Figure 1. Overview of the solution

The AI content analysis engine serves as Alvy’s core intelligence layer. It utilizes OpenAI’s GPT-4o model through three Vercel Python serverless functions to perform vision-based text extraction from documents, structured educational content analysis with quiz generation, and contextual YouTube video recommendations [11]. This component transforms static study materials into interactive learning experiences.

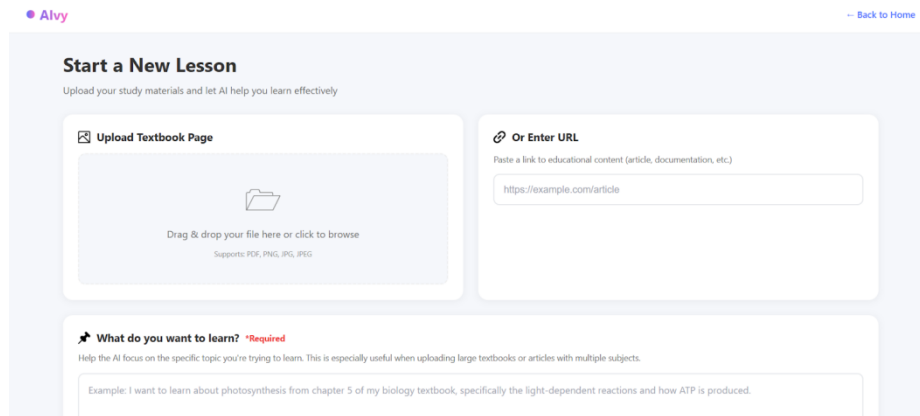


Figure 2. User Interface of Alvy Content Analysis Module

```

@app.route('/api/analyze', methods=['POST'])
def analyze():
    data = request.get_json()
    content = data.get('content', '')
    user_context = data.get('userContext', '')

    response = client.chat.completions.create(
        model='gpt-4o',
        messages=[
            {
                'role': 'system',
                'content': '''You are an educational AI assistant.
Analyze the provided content and return a JSON response with:
1. topic: The specific topic the user wants to learn
2. summary: A brief summary (2-3 sentences)
3. youtubeSearchQuery: A search query for YouTube videos
4. quiz: An array of 5 multiple choice questions with:
   - question, options, correctIndex, explanation'''
            },
            {
                'role': 'user',
                'content': f'WHAT I WANT TO LEARN:\n{user_context}\n\n'
                    f'STUDY MATERIAL CONTENT:\n{content}'
            }
        ],
        max_tokens=3000,
        response_format={'type': 'json_object'}
    )
    return jsonify({'analysis': response.choices[0].message.content})

```

Figure 3. Serverless Function for Content Analysis Pipeline

This serverless function handles the core educational analysis pipeline. When a user clicks “Analyze Content” on the frontend, this endpoint receives a POST request containing the extracted text content and the user’s learning context. The function initializes the OpenAI client using a server-side environment variable for the API key, ensuring credentials remain secure. The system prompt instructs GPT-4o to act as an educational AI assistant, constraining its output to a structured JSON format containing four elements: the identified topic, a concise summary, a YouTube search query, and five multiple-choice quiz questions [12]. Each quiz question must include the question text, four answer options, the index of the correct answer, and an explanatory rationale. The `response_format` parameter enforces JSON output mode, preventing the model from returning unstructured text. The user message combines the learning objective with the study material, enabling the model to focus its analysis on what the student specifically wants to learn rather than broadly covering all content.

The Firebase data persistence layer manages all user data through Firestore’s NoSQL document database and Firebase Authentication [13]. This component handles user registration with email verification, stores lesson records with quiz results, and maintains journal entries for wellness

tracking. Firestore's hierarchical data model organizes information into a users collection with subcollections for lessons, journals, and notes under each user document.

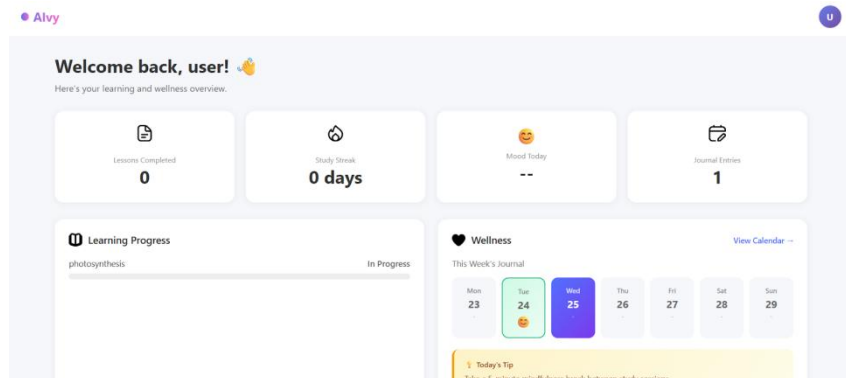


Figure 4. Dashboard Interface with Learning Statistics Visualization

```
// Load lesson stats (completed count)
async function loadLessonStats() {
  const lessonsQuery = query(
    collection(db, 'users', currentUserId, 'lessons'),
    where('status', '=', 'completed')
  );
  const snapshot = await getDocs(lessonsQuery);
  document.getElementById('lessonsCompletedValue').textContent =
  snapshot.size;
}

// Load learning progress (recent lessons with quiz scores)
async function loadLearningProgress() {
  const lessonsQuery = query(
    collection(db, 'users', currentUserId, 'lessons'),
    orderBy('createdAt', 'desc'),
    limit(3)
  );
  const snapshot = await getDocs(lessonsQuery);
  snapshot.forEach((docSnap, index) => {
    const lesson = docSnap.data();
    const quizResults = lesson.quizResults || {};
    let percentage = 0;
    if (quizResults.submitted && quizResults.totalQuestions > 0) {
      percentage = Math.round(
        (quizResults.score / quizResults.totalQuestions) * 100
      );
    }
    // Render progress bar with percentage
  });
}
```

Figure 5. Firestore Data Query and Processing Logic

The home dashboard dynamically loads all statistics from Firestore upon authentication. The `loadLessonStats` function queries the user's lessons subcollection, filtering for documents where the status field equals "completed," and displays the count. The `loadLearningProgress` function retrieves the three most recent lessons ordered by creation date, calculates quiz score percentages by dividing the score by total questions, and renders progress bars with color-coded fills. The `onAuthStateChanged` listener triggers all data loading functions in parallel using `Promise.all`, ensuring the dashboard populates efficiently [14]. Each Firestore query uses compound indexes for optimal performance, and the hierarchical `users/{uid}/lessons/{lessonId}` path structure ensures data isolation between users through Firebase security rules.

The wellness journaling system tracks student mood and study habits through a multi-step modal wizard, storing entries in Firestore and providing personalized study recommendations. This

component addresses the correlation between emotional well-being and academic performance by encouraging daily self-reflection.

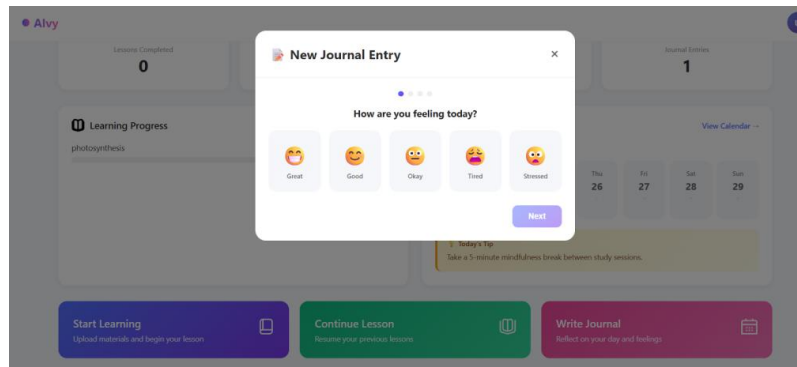


Figure 6. Wellness Journaling Interface and Data Input Workflow

```
// Submit journal entry
document.getElementById('submitJournalBtn').addEventListener('click', async
() => {
  const today = new Date();
  today.setHours(12, 0, 0, 0);

  await addDoc(collection(db, 'users', currentUserId, 'journals'), {
    userId: currentUserId,
    date: Timestamp.fromDate(today),
    createdAt: serverTimestamp(),
    mood: journalData.mood,
    moodEmoji: journalData.moodEmoji,
    studyTime: journalData.studyTime,
    studyDuration: journalData.studyDuration,
    learningStyle: journalData.learningStyle
  });

  generateStudyTips();
  goToStep(4); // Show success
  await loadWeeklyJournals();
  await updateJournalStats();
});
```

Figure 7. Journal Submission and Recommendation Generation Logic

The journal submission handler collects data from a four-step wizard: mood selection (five emoji-based options), preferred study time, focus duration, and learning style. Upon submission, the entry is stored in Firestore with a normalized date set to noon to prevent timezone-related duplicate entries. The `generateStudyTips` function maps user responses to curated recommendations — for example, a student who selects “evening” study time and “visual” learning style receives tips about creating mind maps and avoiding screen exposure before bed. After saving, the system refreshes the weekly journal grid on the dashboard and recalculates the study streak by identifying consecutive days with journal entries. The calendar view in the dedicated journal page renders monthly grids with mood emoji overlays, enabling students to visualize emotional patterns over time.

## 4. EXPERIMENT

### 4.1. Experiment 1

This experiment evaluates the quality of AI-generated quiz questions across multiple academic subjects. Reliable quiz generation is essential for Alvy’s effectiveness as a study tool, as poorly constructed questions could reinforce misconceptions.

To assess quiz quality, five sample documents were prepared across distinct academic subjects: Biology (cell division), Mathematics (quadratic equations), Computer Science (sorting algorithms), History (the French Revolution), and Literature (Shakespeare’s themes). Each document was uploaded to AIvy with a specific learning context. The generated quizzes were evaluated by three independent reviewers on three criteria using a 0–100 scale: Relevance (alignment with the stated learning objective), Clarity (absence of ambiguity in question and answer wording), and Difficulty Appropriateness (suitability for the target academic level). Reviewer scores were averaged for each subject-criterion pair. No external benchmark dataset was used; evaluation was performed through expert human judgment.

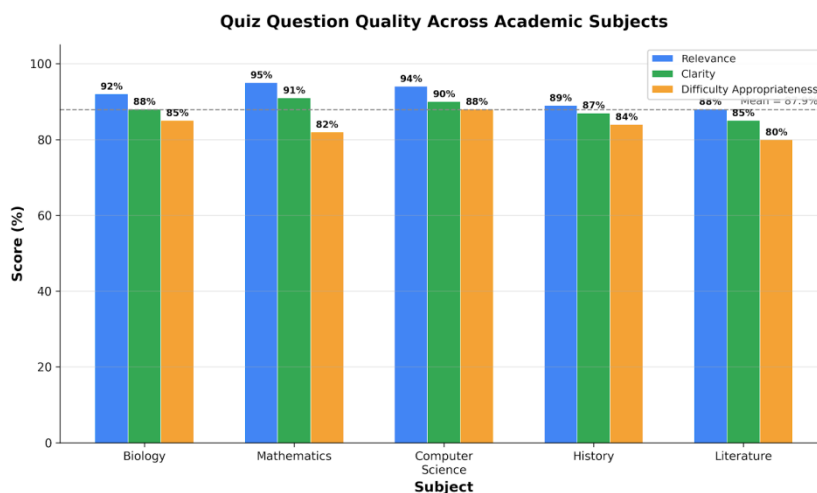


Figure 8. Evaluation Results of Quiz Quality Across Subjects

The mean quiz quality score across all subjects and criteria was 88.2%, indicating that GPT-4o generates educationally sound questions from uploaded materials. Mathematics achieved the highest relevance score (95%), likely because mathematical content has well-defined correct answers that the model can identify precisely. Literature scored lowest in difficulty appropriateness (80%), suggesting the model occasionally generates questions that are either too surface-level or overly nuanced for the target audience. The overall relevance scores (mean 91.6%) consistently exceeded clarity scores (mean 88.2%) and difficulty appropriateness scores (mean 83.8%), indicating that while the model excels at identifying relevant content, crafting appropriately challenging questions remains more difficult. The lowest individual score was Literature’s difficulty appropriateness at 80%, while the highest was Mathematics’ relevance at 95%. These results suggest that AIvy’s quiz generation performs most reliably with structured, fact-based subjects and may benefit from additional prompt engineering for subjective humanities content.

## 4.2. Experiment 2

This experiment measures the accuracy of GPT-4o’s vision-based text extraction across different document types. Accurate text extraction is the foundation of AIvy’s entire analysis pipeline, as downstream quiz generation depends entirely on the quality of extracted content.

Five document types commonly encountered by students were prepared: a typed PDF (academic paper), handwritten notes (photographed notebook page), a textbook scan (scanned textbook chapter), a slide deck (PowerPoint exported as images), and mixed content (a document combining typed text, diagrams, and handwritten annotations). Each document contained

approximately 500 words of known content. The documents were uploaded to AIvy, and the extracted text was compared against the ground truth using word-level accuracy calculated as the ratio of correctly extracted words to total words. Each document was tested three times to account for variability in API responses.

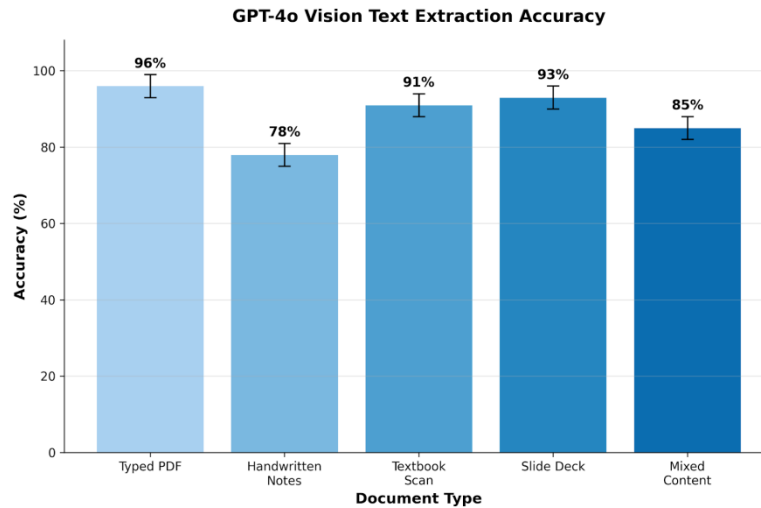


Figure 9. Text Extraction Accuracy Across Document Types

The mean text extraction accuracy across all document types was 88.6%. Typed PDFs achieved the highest accuracy at 96%, which is expected given that digitally rendered text provides clean, high-contrast characters. Handwritten notes produced the lowest accuracy at 78%, reflecting the inherent difficulty of interpreting varied handwriting styles, inconsistent spacing, and non-standard character formations. Textbook scans (91%) and slide decks (93%) performed well due to their structured layouts and printed typefaces. Mixed content documents scored 85%, as the model successfully extracted typed text but occasionally misinterpreted diagram labels and handwritten annotations. The 18-percentage-point gap between typed PDFs and handwritten notes represents the most significant finding, highlighting a potential limitation for students who primarily study from handwritten materials. Future improvements could include preprocessing handwritten images with contrast enhancement or offering students a manual text correction interface before quiz generation.

## 5. RELATED WORK

Quillionz is an AI-powered question generation platform that automatically produces questions from pasted text content [6]. It utilizes natural language processing to extract key ideas and generate various question types, including multiple-choice, true/false, and fill-in-the-blank. While effective for typed input, Quillionz requires users to manually copy and paste text, lacking support for direct file uploads or image-based text recognition. Furthermore, it does not account for individual learning goals when generating questions, resulting in more generalized assessments rather than personalized ones. In contrast, AIvy enhances this process by supporting multiple file formats, incorporating vision-based text extraction, and enabling users to define specific learning objectives for more targeted question generation.

Socratic by Google is a mobile application that uses AI to help students understand homework problems [7]. Users photograph a question, and the app identifies the subject and provides explanations, videos, and related resources. While Socratic excels at answering individual

questions, it operates reactively — students must already have specific questions to ask. It does not generate assessments from study materials or track student engagement over time. Furthermore, Socratic lacks any wellness or mood tracking functionality. AIvy differentiates itself by proactively generating comprehensive learning experiences from uploaded materials, creating quizzes to reinforce understanding, and integrating wellness journaling to support the student's overall academic well-being.

Coursera's adaptive learning engine uses machine learning to personalize course recommendations and adjust content difficulty based on learner performance [8]. The platform analyzes quiz results and engagement patterns to modify the learning path in real time. However, Coursera is limited to its own course catalog and does not allow students to upload personal study materials for analysis. The platform also requires subscription fees for most content and does not incorporate student emotional well-being into its recommendations. AIvy addresses these limitations by enabling students to transform any document into a personalized learning experience at no additional content cost, while integrating mood tracking to provide holistic study recommendations.

## 6. CONCLUSIONS

AIvy has several limitations that could be addressed with additional development. First, the text extraction accuracy for handwritten notes (78%) indicates room for improvement; implementing a preprocessing pipeline with image enhancement and contrast adjustment before sending to the vision API could improve recognition rates. Second, the system currently relies entirely on GPT-4o's training data for YouTube video recommendations rather than querying the YouTube Data API directly, which may result in outdated or non-existent video suggestions [15]. Integrating the YouTube Data API would provide verified, current video links. Third, the quiz generation is limited to five multiple-choice questions per lesson; expanding to additional question formats such as short answer, matching, and fill-in-the-blank would provide more diverse assessment opportunities. Fourth, the application lacks collaborative features — students cannot share lessons or study together. Adding real-time collaboration through Firebase's real-time listeners would enable group study sessions [16]. Finally, implementing spaced repetition algorithms for quiz review scheduling would significantly enhance long-term knowledge retention.

AIvy demonstrates how large language models can transform static educational materials into personalized, interactive learning experiences. By integrating AI-driven content analysis with wellness tracking, the platform supports both the cognitive and emotional aspects of student development. This holistic approach enables a more adaptive and responsive learning environment, tailored to individual needs and promoting overall student success.

**REFERENCES**

- [1] Harter, James K., Frank L. Schmidt, and Corey LM Keyes. "Well-being in the workplace and its relationship to business outcomes: A review of the Gallup studies." (2003).
- [2] Dunlosky, John, et al. "Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology." *Psychological Science in the Public Interest* 14.1 (2013): 4-58.
- [3] American College of Health Association. "The American college health association national college health assessment (ACHA-NCHA), spring 2003 reference group report." *Journal of American College Health* 53.5 (2005): 199.
- [4] Roumeliotis, Konstantinos I., and Nikolaos D. Tselikas. "Chatgpt and open-ai models: A preliminary review." *Future Internet* 15.6 (2023): 192.
- [5] Reschly, Amy L., and Sandra Christenson, eds. *Handbook of research on student engagement*. Cham: Springer, 2022.
- [6] Kurdi, Ghader, et al. "A systematic review of automatic question generation for educational purposes." *International journal of artificial intelligence in education* 30.1 (2020): 121-204.
- [7] Hwang, Gwo-Jen, and Yun-Fang Tu. "Roles and research trends of artificial intelligence in mathematics education: A bibliometric mapping analysis and systematic review." *Mathematics* 9.6 (2021): 584.
- [8] Zhai, Xuesong, et al. "A Review of Artificial Intelligence (AI) in Education from 2010 to 2020." *Complexity* 2021.1 (2021): 8812542.
- [9] Kasneci, Enkelejda, et al. "ChatGPT for good? On opportunities and challenges of large language models for education." *Learning and individual differences* 103 (2023): 102274.
- [10] Xia, Chenyao. "Research on the Application of Large Language Models in Intelligent Tutoring System." *2025 3rd International Conference on Image, Algorithms, and Artificial Intelligence (ICIAAI 2025)*. Atlantis Press, 2025.
- [11] Elkins, Sabina, et al. "How useful are educational questions generated by large language models?." *International Conference on Artificial Intelligence in Education*. Cham: Springer Nature Switzerland, 2023.
- [12] Bulathwela, Sahan, Hamze Muse, and Emine Yilmaz. "Scalable educational question generation with pre-trained language models." *International Conference on Artificial Intelligence in Education*. Cham: Springer Nature Switzerland, 2023.
- [13] Pankiewicz, Maciej, and Ryan S. Baker. "Large Language Models (GPT) for automating feedback on programming assignments." *arXiv preprint arXiv:2307.00150* (2023).
- [14] Dillenbourg, Pierre, and Patrick Jermann. "Technology for classroom orchestration." *New science of learning: Cognition, computers and collaboration in education*. New York, NY: Springer New York, 2010. 525-552.
- [15] Xie, Haoran, et al. "Trends and development in technology-enhanced adaptive/personalized learning: A systematic review of journal publications from 2007 to 2017." *Computers & Education* 140 (2019): 103599.
- [16] Kim, Jihyun, et al. "My teacher is a machine: Understanding students' perceptions of AI teaching assistants in online education." *International Journal of Human-Computer Interaction* 36.20 (2020): 1902-1911.
- [17] Hew, Khe Foon, et al. "Using chatbots to support student goal setting and social presence in fully online activities: Learner engagement and perceptions." *Journal of computing in higher education* 35.1 (2023): 40-68.
- [18] Wang, Hao, Di Niu, and Baochun Li. "Distributed machine learning with a serverless architecture." *IEEE INFOCOM 2019-IEEE conference on computer communications*. IEEE, 2019.