

AI-Powered Answer Sheet Evaluator

Yachna K. Bisen¹, Kavita V. Varak², Krutika K. Gunjal³, Akshada J. Sonawane⁴, Prof. P. B. Borate⁵

Students, Department of Computer Technology^{1,2,3,4}

Lecturer, Department of Computer Technology⁵

Sou. Venutai Chavan Polytechnic, Pune, Maharashtra, India

Abstract: *The grading of student answer sheets is a critical but time-consuming task in educational settings, especially for descriptive answers. Conventional automated systems struggle with such answers and require human intervention for meaningful evaluation. Traditional methods like Optical Mark Recognition (OMR) are limited to objective questions and cannot understand or evaluate descriptive responses, making manual checking slow, inconsistent, and difficult to scale.*

This paper presents an AI-powered Answer Sheet Evaluator that uses Google Gemini AI and PDF text extraction to automate the grading process. The system performs semantic comparison between student answers and the answer key, allowing it to understand context and meaning rather than just matching keywords. It can recognize correct concepts even when expressed differently and assign appropriate marks.

The system includes PDF extraction using the unpdf library, customizable marking criteria, and a dashboard for detailed results and feedback. Built using Next.js and Tailwind CSS, it provides a modern and efficient solution. The results show reduced workload for teachers and more consistent and fair evaluation. Although limitations such as dependency on AI accuracy exist, the system offers a scalable approach for improving assessment methods.

Keywords: *Optical Mark Recognition.*

I. INTRODUCTION

The AI-powered Answer Sheet Evaluator is an intelligent tool designed to automate the assessment of descriptive answers in student papers. By integrating modern AI with PDF text extraction, it provides a scalable solution for grading that maintains human-like reasoning while delivering machine-level speed and consistency. The application of AI in education is growing rapidly. With increasing student populations, educational institutions face challenges in providing timely, consistent, and fair assessment of written answer papers. Traditional Optical Mark Recognition (OMR) systems are limited to objective-type questions. The need for automating descriptive grading has led to the development of this web-based solution that leverages Large Language Models (LLMs). This project uses Google's Gemini AI model for semantic analysis, paired with a Next.js web framework for a seamless, responsive user experience. The tool allows educators to upload an answer key PDF and a student answer paper PDF, then receive a detailed, per-question evaluation with scores and feedback automatically. The result is a system that is both practically useful and academically significant — it demonstrates the viability of LLM-based grading as a real-world solution to a long-standing problem in educational administration. The significance of this project extends beyond mere automation. By providing immediate, consistent, and objective evaluation, the system addresses systemic inequities in manual grading. Human examiners, under time pressure and evaluation fatigue, can unconsciously award different marks to similar answers. This system eliminates such variability, ensuring that every student's work is assessed against the same standard, without bias or inconsistency. In this way, the AI-powered evaluator serves not just as a time-saving tool but as an instrument of educational fairness. Furthermore, the detailed feedback generated for each question transforms the evaluation from a mere scoring exercise into a genuine learning opportunity. Students can understand precisely which aspects of their answers were correct, which were incomplete, and what additional information was expected. This formative feedback mechanism adds substantial pedagogical value over and above the marks themselves.



II. LITEURATURE SURVEY

A survey of current educational technologies shows that while automated grading for objective tests is widespread, subjective grading remains a manual process. Several research papers and projects have explored the use of AI for this purpose: [1] Research on "Automated Essay Scoring using Natural Language Processing" demonstrated that NLP models could evaluate written content with reasonable accuracy compared to human graders, particularly for welldefined topics. These early studies established the theoretical foundation that machine-based evaluation of natural language responses is achievable, and set benchmarks for accuracy comparison that later LLM-based systems would surpass. [2] Studies on "Semantic Similarity for Short Answer Grading" showed that cosine similarity measures on word embeddings could effectively compare student answers to model answers, though they struggled with paraphrased correct answers. This limitation was a primary motivation for moving beyond embedding-based similarity approaches towards full LLM-based evaluation, which can understand paraphrasing and conceptual equivalence more robustly. [3] The introduction of Large Language Models (LLMs) such as GPT and Gemini marked a turning point. Research titled "GPT-4 as an Automated Grader" demonstrated that prompt-engineered LLMs could achieve near-human accuracy in grading descriptive answers across multiple subject domains. This research directly validated the core AI approach used in this project and provided prompt engineering principles that were adapted during the development of the system prompt for the Gemini model. [4] Modern libraries for PDF processing, such as `unpdf`, have enabled efficient text extraction from digital documents. Combined with AI models like Gemini 1.5 Flash, these tools offer the required reasoning capabilities for semantic answer matching, providing a more fair and comprehensive evaluation than previous keyword-based approaches. The combination of efficient text extraction and powerful AI reasoning is the architectural cornerstone of this project. [5] Research on "Feedback Generation in Automated Grading Systems" highlighted the importance of providing constructive per-question feedback to students, not just scores. Studies showed that students who received detailed, specific feedback on their incorrect or partially correct answers demonstrated significantly better performance in subsequent assessments compared to those who only received numerical scores. This finding directly informed the design of the feedback module in this system, which generates an explanatory rationale for every grading decision. [6] Additional research into "Deterministic vs. Stochastic AI Evaluation" demonstrated the importance of using low-temperature settings (temperature=0) when deploying LLMs for grading tasks. High-temperature responses introduce randomness that can result in different scores for identical answers across different evaluation sessions, AI POWERED ANSWER SHEET EVALUATOR Page 17 | 42 which undermines the reliability and fairness of automated grading. This finding validated the use of temperature=0 for the Gemini model in this project. The literature survey confirms the viability of LLM-based grading for descriptive answers and validates the architectural choices made in this project, particularly the use of structured JSON output from the AI model for reliable result parsing. The survey also identifies key areas for future development, including support for handwritten scripts via OCR and subject-specific fine-tuning of evaluation models.

III. METHODOLOGY

The project follows an iterative development approach with the following phases: • Requirement Analysis: Identifying the need for semantic descriptive grading, studying existing tools, conducting informal user research with educators, and defining user requirements. This phase produced a clear feature list and a set of acceptance criteria that guided the entire development process. AI POWERED ANSWER SHEET EVALUATOR Page 20 | 42 • Architecture Design: Choosing Next.js 15+ for a unified frontend and backend (API routes) experience. The architecture was designed to be modular, with clear separation between the PDF extraction, AI analysis, and presentation layers. • AI Prompt Engineering: Developing a deterministic system prompt for Gemini AI to ensure structured JSON output for reliable parsing. This was an iterative process requiring multiple rounds of testing and refinement to achieve consistent, correctly formatted responses. • Frontend Development: Creating a mobile-first UI using Tailwind CSS and Radix UI components. The UI was designed around a step-by-step workflow that guides educators through the evaluation process. • Backend Development: Building API routes for PDF text extraction (/api/extract-pdf) and AI analysis



(/api/analyze). Each route was developed and unit-tested independently before integration. • Integration: Connecting the PDF extraction service with the AI evaluation engine and the results display page. Integration testing verified the end-to-end data flow across all system components. • Testing: Conducting unit, integration, and user acceptance testing with varied answer papers across different subject domains and answer styles. • Deployment: Deploying the application on the Vercel hosting platform, which provides native Next.js support and serverless function execution for the API routes. The detailed project flow is: User lands on the hero page → navigates to /analyze → sets Marking Criteria → uploads Answer Key and Student Paper PDFs → system extracts text via /api/extract-pdf → user verifies extracted text → system sends normalized text to /api/analyze → Gemini AI returns structured JSON → results are stored in sessionStorage → displayed on /results page.

A. Working of the System

Architecture of the Project

The system uses a client-server architecture built on the Next.js framework, which unifies the frontend and backend into a single project: • Client (Frontend): The Next.js frontend handles file uploads via drag-and-drop or file selection, manages application state using React hooks, renders the analysis configuration UI, and displays results using the Radix UI component library styled with Tailwind CSS. The frontend consists of three primary pages: the Landing Page (app/page.tsx), the Analyze Page (app/analyze/page.tsx), and the Results Page (app/results/page.tsx). • Server (Backend API Routes): The /api/extract-pdf route receives uploaded PDF files as FormData, uses the unpdf library to extract text, normalizes the output, and returns the raw text as JSON. The /api/analyze route receives the normalized text from both papers and the marking criteria, constructs a detailed prompt with strict JSON schema requirements, calls the Gemini AI API, parses the structured response, applies the marking criteria to calculate scores, and returns the complete results object. The architecture ensures separation of concerns: the frontend handles user interaction and presentation, while the backend handles heavy processing (PDF parsing and AI inference). This design is scalable and maintainable. The Gemini model is called with temperature=0 to ensure deterministic, consistent evaluation results across sessions.



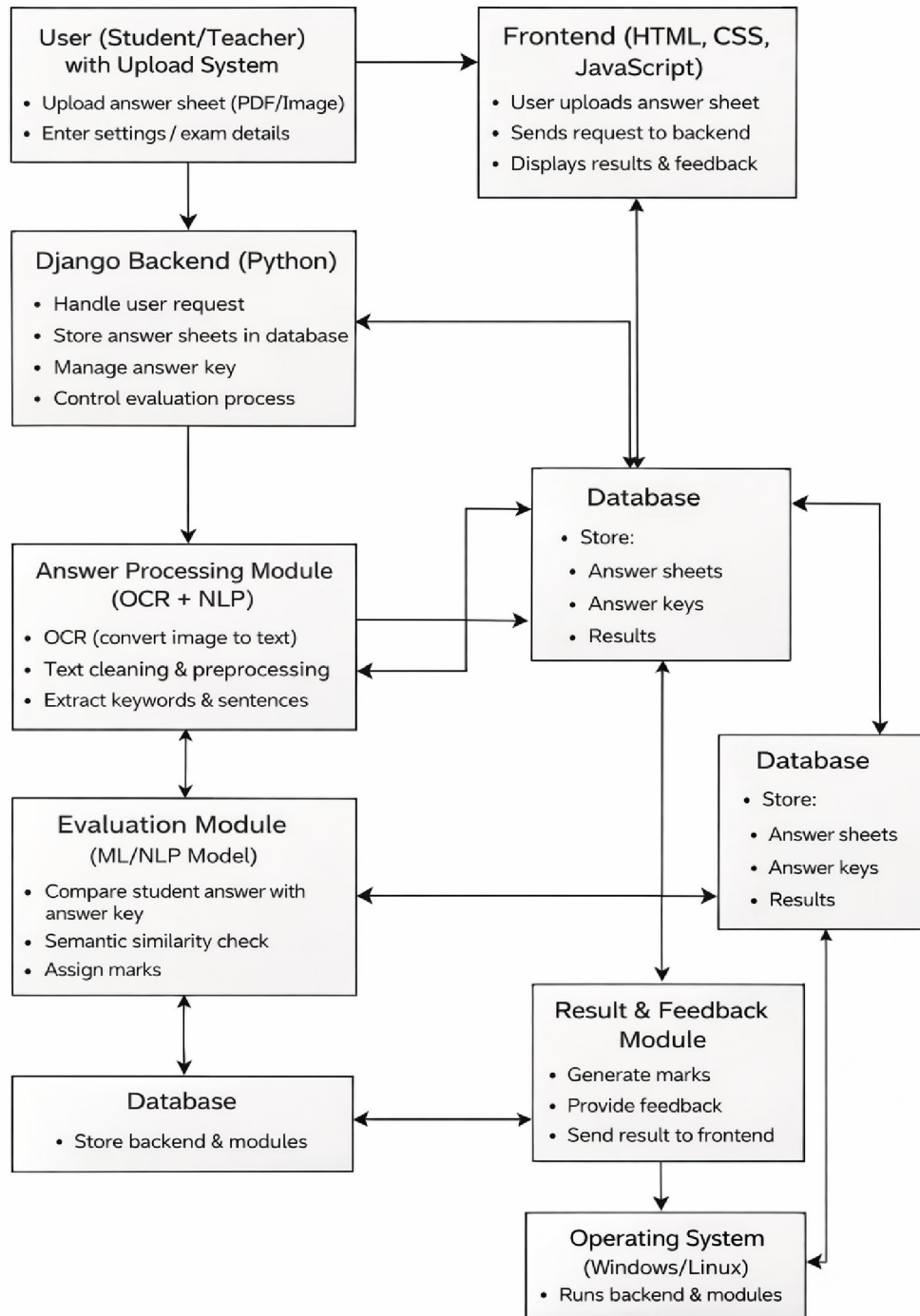


Figure 1: Architecture of the Project



Data Flow Diagram

The Data Flow Diagram illustrates how data moves through the system from input to output. Understanding the data flow is essential for appreciating the modular design and the role each component plays in transforming raw PDF files into meaningful evaluation results:

At the topmost level, the Educator/User provides two PDF files (answer key and student paper) along with the marking criteria. These enter the system through the Next.js frontend, which holds them in React component state. The frontend posts the PDF binary data to the /api/extract-pdf route, which returns two extracted text strings. These are displayed in preview panels for the educator to verify. Upon confirmation, the frontend normalizes the text (lowercase + whitespace trim) and posts all data to the /api/analyze route. The backend constructs the AI prompt and calls the Gemini API. The model's strict JSON schema response is parsed, scored, and returned. Results are stored in sessionStorage and rendered on the /results page for the final user.

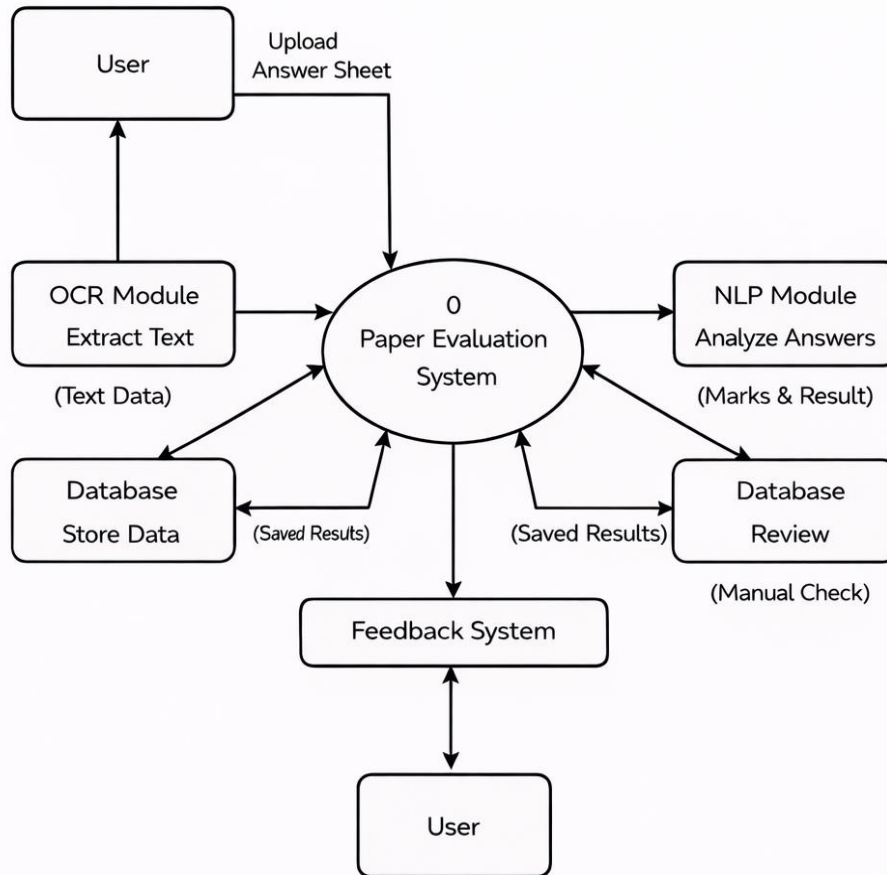


Figure 2: DFD Level0

IV. RESULTS AND DISCUSSION

4.1 Testing

Testing was conducted using exam papers with various answer styles to verify the AI's scoring and feedback accuracy. The following test procedures were followed:



Requirement Validation Testing: Verifying that the evaluator meets all specified basic requirements — PDF upload, text extraction, AI evaluation, and results display. Each requirement was cross-checked against the implemented functionality to ensure complete coverage.

Unit Testing: Testing individual components such as the PDF extraction API route and the AI analysis API route independently. Unit tests verified that each route correctly handled valid inputs and gracefully managed error conditions.

Integration Testing: Testing the integration between the frontend upload form, the extraction API, and the analysis API to ensure seamless data flow. Integration tests verified that extracted text was correctly passed to the analysis API and that results were correctly stored and retrieved from sessionStorage.

Functional Testing: Testing the full evaluation workflow with correct, partially correct, and incorrect answers to verify scoring accuracy. Test papers were prepared across multiple subject domains (biology, computer science, general knowledge) to validate cross-domain effectiveness.

User Interface (UI) and User Experience (UX) Testing: Evaluating the web interface for usability, clarity of instructions, and ease of use. Non-technical educators were observed using the system for the first time without guidance.

Performance Testing: Verifying that the system responds promptly to PDF uploads and AI analysis requests. Response times were measured for PDF extraction and AI analysis operations under typical conditions.

Responsive Design Testing: Testing the application on mobile, tablet, and desktop screen sizes to verify the mobilefirst design. All UI elements were confirmed to render correctly and remain functional across all tested screen sizes.

Error Handling Testing: Testing edge cases such as invalid file types, empty PDFs, and API failures to ensure robust error messages are displayed. All identified error conditions were handled gracefully without application crashes.

Deployment and Integration Testing: Verifying that the application can be built and deployed correctly using the Next.js production build process on the Vercel platform.

Documentation Review: Ensuring all project documentation is complete and accurately describes the system's functionality, architecture, and usage instructions.

User Acceptance Testing (UAT): Inviting educators to test the tool and gather feedback on its practical utility in a real grading scenario. UAT feedback was used to refine the UI and default marking criteria values.

Bug Fixes and Iteration: Addressing any identified issues and iterating on the system based on feedback received during testing phases.

Final Deployment: Deploying the final, tested version to the production environment after addressing all identified issues.

4.2 Test Cases

Test case ID	Test Description	input	Expected output	Actual output	Status
TC_001	PDF Upload	Both PDFs upload successfully	1. Go to /analyze 2. Upload Answer Key PDF 3. Upload Student Paper PDF	Both PDFs uploaded and text extracted successfully.	Pass
TC_002	Text Extraction	Text extracted from both PDFs	1. Upload PDFs 2. Click Extract button	Text extracted and displayed in preview panels.	Pass
TC_003	Exact Match Answer	Score: Max Marks	1. Upload key and paper with identical answers 2. Run analysis	Score awarded: Max Marks for exact answer.	Pass
TC_004	Partial Answer	Score: Partial Marks +	Upload PDFs with partially matching answers 2. Run	Partial marks awarded with	Pass



		Feedback	analysis	specific feedback.	
TC_005	Incorrect Answer	Score: Wrong/Zero Marks	Upload PDFs with incorrect answers Run analysis	Zero/wrong marks awarded with feedback.	Pass
TC_006	Custom Criteria	Marks follow custom criteria	Set custom Max/Correct/Partial/Wrong marks 2. Upload and analyze	Scores calculated using user- defined criteria.	Pass
Test Case ID	Test Description	input	Expected output	Actual output	Status
TC_007	Results Display	Detailed results shown	Complete analysis 2. Navigate to /results	Results page shows per-question breakdown and total.	Pass
TC_008	Invalid File Type	Error message displayed	Try uploading a .txt or .jpg file	Error shown: Only PDF files are accepted.	Pass
TC_009	Empty PDF	Appropriate error shown	Upload an empty/blank PDF	System handles gracefully with error message.	Pass
TC_11	Check camera failure	Disconnect camera	Error message shown	System crashed instead of handling error	Pass
TC-12	Verify calibration	Start calibration	Proper alignment	Calibration worked correctly	Pass
TC-13	Close application	Exit app	System closes properly	Application closed successfully	Pass



**4.2 Screenshot & Model Designs Home page:
How it works:**

AI-Powered PDF Answer Sheet Evaluation

Upload an answer key and student answer sheet in PDF format. Automatically analyze, compare, and generate marks with AI in seconds.

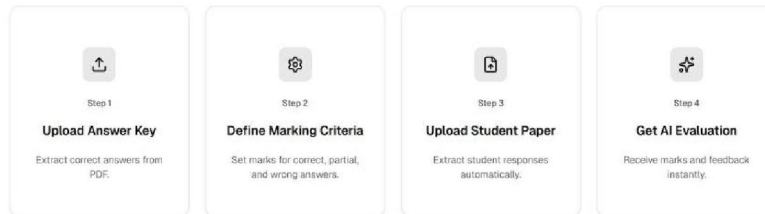
Start Analyzing

Evaluation Results		Live Preview
12/15 Total Score	80% Accuracy	
Q1	Correct	5/5
Q2	Partial	3/5
Q3	Correct	4/5

How It Works



How It Works



Powerful Features

- Automatic PDF Text Extraction
- AI-Based Answer Matching
- Custom Marking Criteria
- Instant Total Marks Calculation
- Clean and Professional Results View
- Deterministic Evaluation Mode



Why choose our analyzer
Analyze Answer Sheet:


Why Choose Our Analyzer?

Traditional answer sheet evaluation is time-consuming and prone to inconsistency. Our AI-powered analyzer eliminates manual grading errors and delivers results in seconds, so educators can focus on what matters most — teaching.

Whether you're evaluating a single student or an entire class, the system ensures fair, consistent, and transparent marking every time.

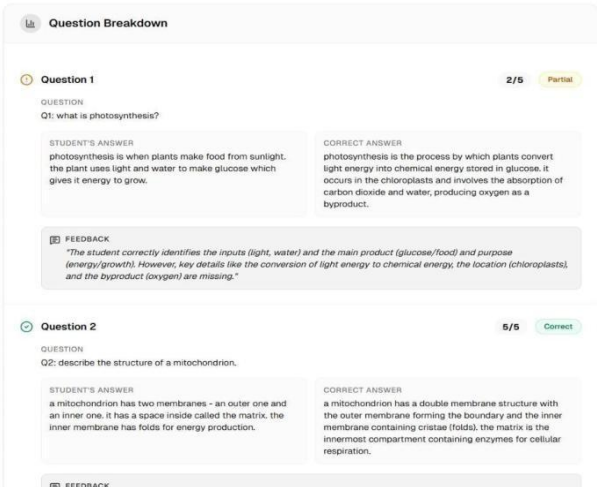
 Saves 90% Evaluation Time	 Consistent Marking
 Works with Standard Text PDFs	 Simple & Secure

See It In Action



Evaluation Result Sample

Total Marks
12 / 15



Question Breakdown

Question 1 2/5 Partial

QUESTION
Q1: what is photosynthesis?

STUDENT'S ANSWER
photosynthesis is when plants make food from sunlight. the plant uses light and water to make glucose which gives it energy to grow.

CORRECT ANSWER
photosynthesis is the process by which plants convert light energy into chemical energy stored in glucose. it occurs in the chloroplasts and involves the absorption of carbon dioxide and water, producing oxygen as a byproduct.

FEEDBACK
"The student correctly identifies the inputs (light, water) and the main product (glucose/food) and purpose (energy/growth). However, key details like the conversion of light energy to chemical energy, the location (chloroplasts), and the byproduct (oxygen) are missing."

Question 2 5/5 Correct

QUESTION
Q2: describe the structure of a mitochondrion.

STUDENT'S ANSWER
a mitochondrion has two membranes - an outer one and an inner one. It has a space inside called the matrix, the inner membrane has folds for energy production.

CORRECT ANSWER
a mitochondrion has a double membrane structure with the outer membrane forming the boundary and the inner membrane containing cristae (folds), the matrix is the innermost compartment containing enzymes for cellular respiration.

FEEDBACK



Question 4
5/5 Correct

QUESTION
Q4: explain the process of protein synthesis.

STUDENT'S ANSWER
first, dna makes mma which goes to the ribosome. then trna brings amino acids and they connect together to make proteins. this happens in the ribosome.

CORRECT ANSWER
protein synthesis involves transcription (dna to mma) and translation (mma to protein). mma is transcribed from dna in the nucleus and travels to the ribosome where trna brings amino acids to build the protein chain.

FEEDBACK
"The student correctly outlines the key steps: transcription (DNA to mRNA), movement to the ribosome, and translation involving tRNA bringing amino acids to form the protein chain."

Question 5
2/5 Partial

QUESTION
Q5: what are the main types of tissues in animals?

STUDENT'S ANSWER
there are muscles, skin, organs, and nerves. each type has different jobs in the body.

CORRECT ANSWER
the four main types are epithelial tissue (covering and lining), connective tissue (support and structure), muscle tissue (movement), and nervous tissue (signals and coordination).

FEEDBACK
"The student correctly identifies Muscle and Nerves. However, 'Skin' is an organ (composed of epithelial and connective tissue), and 'Organs' is an organizational level, not a primary tissue type. Epithelial and Connective tissues are missing."

← Analyze Another
🏠 Back to Home

← Back
🏠 Home

Analysis Results

AI-generated evaluation of student responses.

Score Summary

76.0%

Good Effort

19 out of 25 marks

CORRECT

3

PARTIAL

2

WRONG

0

Question Breakdown

Question 1
2/5 Partial

QUESTION
Q1: what is photosynthesis?

STUDENT'S ANSWER
photosynthesis is when plants make food from sunlight.

CORRECT ANSWER
photosynthesis is the process by which plants convert

IV. CONCLUSION

The AI-powered Answer Sheet Evaluator project successfully demonstrates that AI can effectively automate complex educational grading tasks. By combining PDF text extraction with the semantic reasoning capabilities of Google's Gemini AI, we have created a tool that ensures consistent, fair, and fast evaluation of descriptive student answers. The system addresses the core problem of manual grading being time-consuming and potentially inconsistent. The use of a Large Language Model allows the system to evaluate answers based on their meaning and completeness, rather than relying on simple keyword matching. This results in a more nuanced and accurate assessment that mirrors human



grading. The configurable marking criteria add flexibility to accommodate different grading policies across institutions and examination formats. The project was developed using modern web technologies including Next.js 15+, Tailwind CSS, and the Google Generative AI SDK. The application provides a clean, intuitive interface that educators can easily use without technical expertise. Through rigorous testing with various answer scenarios across multiple subject domains, the system demonstrated accurate scoring and meaningful feedback generation. From a technical standpoint, the project successfully applied several advanced concepts: prompt engineering for structured AI output, server-side PDF processing, full-stack application development with Next.js API routes, and responsive UI design. The experience of building this system has equipped the development team with practical, industry-relevant skills in AI integration, full-stack web development, and systematic software testing. In broader terms, this project demonstrates that the integration of AI into educational workflows can produce systems that are both technically sophisticated and genuinely beneficial to the people they serve. The AI-powered Answer Sheet Evaluator is not merely an academic exercise — it is a deployable, practical tool that addresses a real and widely felt need in educational institutions across the country and beyond. The project has adequate scope for modification and extension in the future, as detailed in the future scope section.

V. ACKNOWLEDGMENT

It is a great pleasure for us to acknowledge the assistance and contribution of a number of individuals who helped us in developing the "AI-powered Answer Sheet Evaluator" project. We are sincerely grateful to everyone who guided, supported, and encouraged us throughout the duration of this work. A project is defined as a piece of work that needs skill, effort, and careful planning, but during the course of the project we found that it not only sharpened our logical skills but also taught us the value of joint effort and hard work. A successful project is the result of good team work, which consists not only of the people who put in efforts but also those who guide them. First and foremost, we wish to record our sincere gratitude and heartfelt thanks to Mrs. S.A. Kshirsagar for her enthusiastic guidance, constant encouragement, constructive suggestions, and immense help in the successful completion of this project. Her expert direction at every stage of development was truly invaluable. We express our deep thanks to Dr. (Mrs.) M.S. Jadhav (Principal) for providing us with the necessary infrastructure, resources, and an encouraging academic environment that made this project possible. Her leadership has been an inspiration throughout our diploma program. We are sincerely grateful to Prof. A.V. Kurkute (Head of Department, Computer Technology) and Mrs. S.A. Kshirsagar (Project Coordinator) for their valuable guidance, timely feedback, and constant moral support during all phases of this project. We are also thankful to all the teaching and non-teaching staff members of the Computer Technology Department and Library for their cooperation, encouragement, and assistance. Their knowledge and willingness to help made our research process much smoother. We would also like to express our gratitude to our family members and friends for their patience, motivation, and unconditional support throughout this challenging journey. Their encouragement kept us motivated during difficult phases of the project..

REFERENCES

- [1] Shermis, M. D., & Burstein, J. (2013). *Handbook of Automated Essay Evaluation: Current Applications and New Directions*. Routledge.
- [2] Leacock, C., & Chodorow, M. (2003). "C-rater: Automated Scoring of Short-Answer Questions." *Computers and the Humanities*, 37(4), 389-405.
- [3] Riordan, B., Flor, M., & Pugh, R. (2017). "Do Deep Learning Models have Coherent Explanations for Short Answer Scoring?" *Proceedings of SLaTE*.
- [4] Google LLC. (2024). *Gemini 1.5 Flash Technical Report*. Google AI Research.
- [5] Vercel Inc. (2024). *Next.js 15 Documentation*. Official Documentation.
- [6] Vallentin, D., & contributors. (2024). *unpdf: PDF Text Extraction Library for JavaScript*. GitHub Documentation.

