MOBILE AND SIMULATION-BASED APPROACH TO REDUCE DYSLEXIA IN CHILDREN

Shangeeth. V - IT21273858

Final Report

Bachelor of Science (Hons) Degree in Information Technology, Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

MOBILE AND SIMULATION-BASED APPROACH TO REDUCE DYSLEXIA IN CHILDREN

Shangeeth. V - IT21273858

Final Report

Bachelor of Science (Hons) Degree in Information Technology, Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

April 2025

DECLARATION

I declare that this is my own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously publish or written by another person expect where the acknowledgement is made in the text.

Name	Student ID	Signature
Shangeeth. V	IT21273858	V. Shangeeth

The supervisor/s should certify the proposal report with the following declaration.
The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

…………………………
Signature of the Supervisor
(Dr. Mr. Kapila Dissanayaka)

…………………
Date

………………………………
Signature of the Co-Supervisor
(Ms. Rivoni De Zoysa)

…………………
Date

ABSTRACT

This study aims to create a smart and customized educational game distribution system designed for children with dyslexia, emphasizing the improvement of reading abilities via engaging gameplay and flexible learning models. The initiative tackles the various cognitive and behavioral difficulties encountered by dyslexic students by incorporating a machine learning-driven game suggestion system and a real-time performance assessment mechanism. By employing deep learning algorithms, the system categorizes and creates visual content like shapes and patterns, essential elements in visual learning therapy. [1] [2]

The fundamental approach encompasses training convolutional neural networks (CNNs) for shape identification and sequence forecasting by employing LSTM (Long Short-Term Memory) models to examine time-dependent patterns in gameplay actions. An additional GAN (Generative Adversarial Network) was investigated for generating colorful shape images, but it was subsequently deprecated because of excessive resource consumption and performance variability. To guarantee reliability and prevent overfitting, methods like dropout layers, data augmentation, and validation checks were utilized during the training of the model. The trained models are provided through a Python-based Flask API that works smoothly with a Node.js backend, which handles user profile management, score tracking, and game assignments. [3] [4] [5] [6] The system's smartness is additionally improved by a scoring system that changes dynamically according to the game result (win or lose) and completion time. Children who regularly excel are slowly given tougher games, maintaining ongoing involvement and advancement. The backend functionality, created with Node.js and Prisma ORM utilizing MongoDB, guarantees effective data handling and growth potential [6] [7]

The system integrates computer vision and AI with an online educational platform, improving personalized learning environments and scalability for dyslexic children's educators. Its modular structure supports future improvements, promoting accessibility and creativity in assistive learning tools.

Keyword: - Dyslexia, Educational Games, Artificial Intelligence, Shape Recognition, Pattern Prediction, Deep Learning, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Adaptive Learning, Flask API, Node.js Backend, MongoDB, Visual Skill Enhancement, Gamification, Machine Learning in Education.

ACKNOWLEDGEMENT

First and foremost, we want to convey our heartfelt thanks to our supervisor, **Dr. Mr. Kapila Dissanayaka**, for his ongoing guidance, invaluable insights, and steadfast support during this research period. His specialized knowledge and helpful critiques have been essential in influencing the course and achievement of this project.

We are genuinely grateful to our co-supervisor, **Ms. Rivoni De Zoysa**, for her unwavering support, valuable insights, and prompt feedback that significantly improved the quality of our work.

We express our sincere gratitude to our external supervisor, **Dr. P. Gowritharan**, whose insightful feedback and critical assessments aided us in upholding academic rigor and practical relevance in our research results.

We also want to express our gratitude to the faculty members and colleagues at our university who offered us moral encouragement, resources, and collaborative inspiration during difficult periods.

In conclusion, we appreciate our families and friends for their support, patience, and encouragement during the duration of this project. Their unwavering support provided us with the strength to surmount challenges and stay focused on our objectives.

This study could not have been achieved without the combined efforts of all these people, to whom we are genuinely grateful.

TABLE OF CONTENTS

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Full Term
AI	Artificial Intelligence
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
ML	Machine Learning
API	Application Programming Interface
UI	User Interface
UX	User Experience
NLP	Natural Language Processing
OCR	Optical Character Recognition
ORM	Object Relational Mapping
JWT	JSON Web Token
GPU	Graphics Processing Unit
JSON	JavaScript Object Notation
HTTP	HyperText Transfer Protocol
CRUD	Create, Read, Update, Delete
DB	Database
AWS	Amazon Web Services
PWA	Progressive Web Application
CSV	Comma-Separated Values
IDE	Integrated Development Environment
SSD	Solid State Drive
I/O	Input/Output
UAT	User Acceptance Testing
MVC	Model-View-Controller
UXD	User Experience Design
SEO	Search Engine Optimization
SDK	Software Development Kit
MVP	Minimum Viable Product
GAN	Generative Adversarial Network
FOSS	Free and Open-Source Software
HTTPS	Hypertext Transfer Protocol Secure
Colab	Google Colaboratory
DPI	Dots Per Inch
STT	Speech-to-Text
TTS	Text-to-Speech
DBMS	Database Management System
NoSQL	Non-relational Structured Query Language Database
DNS	Domain Name System

Table 1 List of Abbreviation

1. INTRODUCTION

1.1. Background Study and Literature Review

1.1.1 Background Study

The growing focus on tailored education, particularly for kids facing learning challenges like dyslexia, has resulted in major advancements in adaptive learning technologies. Among these, educational games have been shown to effectively boost engagement and learning results by customizing content to meet the specific needs of individual learners. In this setting, the combination of artificial intelligence (AI) with educational systems provides an effective method to automate customization and monitor student advancement instantly. [7]

Many research studies have emphasized the significance of early interventions for children with dyslexia, indicating that visual and pattern-based learning resources demonstrate considerable potential. Conventional educational games typically depend on fixed difficulty levels and broad feedback, which might not address the specific requirements of individual learners. Consequently, there is an increasing demand for adaptive systems that can modify content according to the learner's existing skill level, advancement, and engagement statistics.

Recent progress in machine learning, especially deep learning architectures like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), has facilitated precise forecasting and categorization of visual data and sequences. These models have been successfully utilized in tasks related to image recognition like recognizing geometric shapes and also in challenges involving pattern prediction. Utilizing these models enables us to evaluate a child's comprehension of visual materials in real-time and adjust the game's difficulty as needed. [3] [9]

To avoid overfitting and promote generalization across various inputs, methods like dropout, data augmentation, and batch normalization are commonly utilized during training stages. These methods enhance model robustness, particularly when the datasets are either small or generated synthetically.

Regarding backend infrastructure, Node.js has gained popularity for creating scalable server-side applications because of its event-driven architecture and asynchronous I/O management. At the same time, Python continues to be a dominant language for the development and processing of AI models because of its vast library support and community. To connect the two environments, Flask is commonly utilized to develop lightweight RESTful APIs in Python that provide predictions or handle data, which can then be used by the Node.js backend.

NoSQL databases like MongoDB provide schema flexibility and quick access speeds for data storage and retrieval, making them ideal for applications where user profiles, game statuses, and analytics require frequent updates and queries.

In this project, we investigated a system that utilizes AI models to allocate educational games according to a child's existing skill level, assesses their performance, and modifies their learning path. Models were developed to categorize shapes, foresee visual patterns, and additionally create synthetic images of shapes. Via real-time API requests, the system assesses input images or sequences and adjusts the child's game level accordingly. A performance accuracy and completion time-based point system offers enhanced personalization and motivation.

This approach seeks to develop a more interactive and impactful educational experience for kids with dyslexia, enhanced by smart automation and ongoing progress tracking.

1.1.2 Literature Review

The increasing demand for tailored learning systems, particularly for kids with learning disabilities such as dyslexia, has driven significant research in artificial intelligence (AI), educational gaming, and adaptive learning technologies. This review of literature centers on previous research and methods pertaining to shape identification, pattern forecasting, adaptive learning games, and the incorporation of machine learning into child-focused educational systems.

Educational Games and Cognitive Development:

Educational games have been recognized for a long time as impactful resources in improving cognitive and motor abilities in children. Papastergiou (2009) states that environments centered around game-based learning can engage students and enhance their problem-solving abilities by providing instant feedback and engaging visual elements. Additionally, Prensky (2001) highlighted that digital game-based learning can connect entertainment with education, making it especially impactful for youthful learners. For kids facing learning challenges like dyslexia, games offer a beneficial and interactive way to enhance visual and memory abilities (Shalev et al., 2007).

Shape and Pattern Recognition Using CNN and RNN Models:

Convolutional Neural Networks (CNNs) are extensively utilized in image classification, especially in scenarios related to shape detection. Lecun et al. (1998) presented CNNs as powerful methods for pattern recognition because of their hierarchical learning and spatial invariance. This is especially pertinent for educational tools that depend on recognizing geometric shapes from user contributions. In connected research, Suk et al. (2016) showed that CNNs can be utilized for real-time classification of objects in visual learning environments. [3]

In contrast, pattern prediction is more suited for temporal sequence data and is typically handled with Recurrent Neural Networks (RNNs) or Long Short-Term Memory networks (LSTMs). Graves et al. (2013) effectively utilized LSTM networks for sequence prediction challenges, demonstrating their ability to learn from time-related dependencies. [9] [6]These models have been utilized for predicting learning progress and generating adaptive content within intelligent tutoring systems (Piech et al., 2015).

Adaptive Learning Systems:

Adaptive learning systems change the content or challenge level in real-time according to the learner’s advancement. Brusilovsky and Millán (2007) defined adaptive systems as ones that can customize educational content to fit a learner’s strengths and weaknesses. Recent research by Khosravi et al. (2017) highlighted the role of predictive analytics in adaptive educational technologies to enhance personalization and student outcomes.

Overfitting and Generalization Techniques:

Overfitting presents a significant challenge when training deep learning models with small datasets. Goodfellow et al. (2016) explored different methods to reduce overfitting, including dropout, batch normalization, and data augmentation. These methods are crucial for creating strong models that perform effectively on new, unfamiliar data especially when working with synthetic or constrained real-world datasets, like those employed in shape recognition activities for children.

System Integration with Node.js, Flask, and MongoDB:

Incorporating AI models into real-time educational platforms necessitates strong backend support. Node.js is renowned for its non-blocking I/O architecture and scalability, which makes it ideal for managing several game sessions and user requests simultaneously (Tilkov & Vinoski, 2010). Flask, a minimalist Python web framework, is frequently utilized to present AI model predictions as RESTful APIs. This allows for smooth interaction between the AI modules and the application interface. MongoDB, a NoSQL document-oriented database, is ideal for handling dynamic user profiles, gaming history, and scoring information because of its flexibility and superior performance (Chodorow, 2013). [6] [7]

Summary of Gaps and Contributions:

Although many studies have investigated adaptive learning and AI in educational settings, only a small number have concentrated specifically on children with dyslexia and their distinct requirements in visual learning. Furthermore, the incorporation of real-time performance monitoring, dynamic game allocation, and AI-driven visual forecasting is still not well investigated. This study tackles these deficiencies by creating a game task engine that utilizes CNN and LSTM models for shape identification and pattern forecasting, respectively, and employs performance-driven assessment to modify game difficulty. The suggested system improves learning by making instant decisions based on the player's advancement, precision, and pace. [6] [3]

1.2. Research Gap

Despite the notable progress in educational technologies and machine learning-based learning platforms, there persists a crucial gap in the successful implementation of individualized visual learning systems for children with particular learning challenges, especially dyslexia. Many research efforts have recognized the cognitive advantages of educational games; however, only a limited number of systems are specifically created to adjust dynamically according to the performance of each learner. The majority of traditional game-based learning platforms provide a fixed difficulty level and do not customize game tasks based on a learner's existing skills or advancements over time.

Additionally, although convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have shown remarkable effectiveness in image classification and sequence prediction respectively, their actual incorporation into real-time educational systems is constrained. Current applications either emphasize predictive models independently or do not have a strong backend to facilitate adaptive learning paths driven by predictive feedback. The capacity to connect these models with real-time databases and backend systems for seamless user experience is still not fully investigated. [3] [9]

Another significant constraint in existing systems is the lack of smart assessment methods that take into account both precision and completion duration when determining learning levels or rewards. Numerous systems evaluate performance solely on accuracy, overlooking other important factors like speed, reliability, and the learner's past development. This results in a disparity in equitable and thorough performance assessment, especially for students who might need additional time because of learning difficulties.

Additionally, many relevant studies do not implement strategies to address overfitting in limited or artificial datasets, a frequent issue in specialized educational contexts. In the absence of effective generalization techniques such as data augmentation, dropout, or batch normalization, the model's effectiveness in practical situations declines markedly.

This study tackles these shortcomings by suggesting an adaptive educational game engine that combines CNN and LSTM models with a backend built on Node.js and Flask. It features smart performance monitoring, instant game assignment according to difficulty, and analytics-based learning advancement. The implementation of overfitting control strategies, combined with a dynamic scoring system, guarantees improved personalization accuracy. Consequently, this system aids in creating a more adaptable and student-centered educational platform, particularly for children needing customized cognitive assistance. [3] [6]

1.3. Research Problem

In the current fast-changing educational environment, incorporating technology into personalized learning has become more crucial than ever. Nonetheless, kids with learning challenges like dyslexia remain inadequately supported by traditional e-learning platforms. These platforms typically lack design to cater to the varied cognitive skills of learners and do not adjust in real time according to user performance. While there are educational games designed to improve engagement and understanding, many of them do not possess the intelligence necessary to tailor the learning experience effectively. As a result, students especially young ones, might either get frustrated with challenges that are excessively difficult or lose interest in activities that are overly simple.

Current digital learning platforms usually function on a set progression model, requiring all learners to adhere to a specific path without regard to their skills or outcomes. This uniform approach fails to recognize the need to dynamically modify the complexity and type of learning material according to the learner's response. Especially for kids who have difficulties with visual processing and recognizing patterns crucial cognitive elements impacted by dyslexia such unchanging systems fall short.

Even though machine learning is increasingly utilized in education, the majority of applications are confined to classification activities, like sorting test responses or forecasting student dropouts. These algorithms are seldom used for real-time decision-making in an interactive learning setting. Moreover, the absence of smooth integration among frontend interfaces, backend systems, and predictive models leads to disjointed learning experiences. Although models such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks excel in image recognition and pattern prediction, their capabilities are mostly underutilized in adaptive educational gaming, particularly in resource-limited, real-time settings. [6] [3]

Another significant issue in existing systems is their failure to assess students comprehensively. Many depend exclusively on assessments with right-or-wrong answers, entirely overlooking other aspects like the duration taken to finish a task or the learner's historical performance patterns. As a result, this leads to a skewed and frequently unjust evaluation of a student's true comprehension and development. Children experiencing dyslexia, for example, might arrive at correct solutions but need additional time because of slower visual processing. Punishing them for taking extra time not only impacts their self-esteem but also distorts performance metrics, undermining the effectiveness of tailored learning paths.

Compounding this complexity is the technical challenge of overfitting when developing models using restricted or synthetic datasets. Since personalized learning systems frequently function in specialized areas with limited data access, the danger of overfitting becomes paramount. In the absence of adequate generalization methods like dropout regularization, data augmentation, and batch normalization the models struggle to deliver consistent performance in real-world situations. This impacts not just model precision but also diminishes trust in the system's suggestions.

In addition, many adaptive learning systems lack cross-platform compatibility. They do not have the backend capabilities to handle user profiles, authentication, data storage, and real-time analytics. There is also a lack of well-organized API ecosystems that facilitate the easy incorporation of machine learning outputs into user interfaces. This restricts the practicality and expandability of these systems in actual educational environments, particularly for young users who need continuous supervision and assistance.

This study aims to tackle these challenges by suggesting a cohesive system that integrates machine learning, smart game allocation, and real-time performance monitoring. The suggested approach combines a CNN model to forecast shapes from image inputs, an LSTM model to anticipate the following shape in a sequence, and a Generative Adversarial Network (GAN) to produce visually detailed images of shapes for training and gaming purposes. These models are coordinated via a Flask-powered Python API that connects effortlessly with a Node.js backend, handling user information, authentication, and game logic with MongoDB and Prisma ORM. [7] [6]

The system also incorporates a game execution engine that assesses the learner's performance based on accuracy, time spent, and progression history. This adaptive scoring system modifies the learner's level and assigns games accordingly. For instance, a child who reliably excels in 'easy' level games over brief periods will advance to 'medium' level challenges. Conversely, if a child has difficulty with 'medium' level tasks and often loses, they might be switched to 'easy' level content to regain confidence.

By means of real-time adaptability, the suggested solution responds to the main research issue: the absence of smart, tailored, and accessible visual learning settings for children facing cognitive learning difficulties. It maintains the proper equilibrium between challenge and support, enhancing the learning experience while sustaining user motivation and involvement. The model integrates optimal strategies to reduce overfitting and enhance inference accuracy, resulting in a strong educational resource suitable for practical application.

1.4. Research Objectives

1.4.1. Main objectives

The primary goal of this study is to create and implement an educational game-based system integrated with AI, specifically designed for children with dyslexia, aimed at improving their cognitive and reading skills. The system seeks to offer a tailored and flexible learning experience by utilizing smart gameplay features, immediate evaluation, and automatic feedback. By combining machine learning models with game-oriented teaching methods, the initiative aims to close the divide between classic interventions and contemporary digital solutions, providing a scalable, interactive, and approachable platform that fosters ongoing enhancement and data-informed decision-making for learners, educators, and guardians. [11]

1.4.2. Specific objective

- To create a modular game framework that focuses on various skills such as visual processing, shape recognition, and pattern identification, essential for reading growth in children with dyslexia.
- To develop and assess AI/ML models (e.g., CNN for shape recognition, LSTM for sequence forecasting) with a tailored dataset for precision and responsiveness. [12] [6] [13] [3]
- To create an intelligent backend that flexibly allocates games according to the child's present level, historical performance, and reaction time, guaranteeing a personalized learning journey.
- To link the game engine to a backend developed in Node.js and Python (Flask) via RESTful APIs, guaranteeing seamless communication and immediate feedback. [14] [4]

To efficiently store and manage progress data utilizing MongoDB with Prisma ORM, enabling analysis and reporting. [6] [7]

1.4.3. Business objective

1. **Market a Scalable Educational Platform:**

   The platform aims to meet the increasing need for gamified educational resources in the special education field. As awareness of neurodiversity and inclusive education grows, this solution can be marketed as a tailored product for schools, therapists, and parents. Its modular structure enables expansion for additional learning disorders beyond dyslexia later on.

2. **Introduce Intelligent Personalization in EdTech:**

   In contrast to standard game-based applications, our platform leverages AI models to customize content dynamically. This guarantees that each child receives the appropriate challenge level and feedback, enhancing their engagement and results. This feature enhances the platform's appeal to B2B customers in the education and healthcare industries who prioritize data-driven, adaptive learning.

3. **Data-Driven Reporting for Stakeholders:**

   The combination of cloud storage and analytics facilitates thorough reporting for educators, therapists, and parents. Stakeholders are able to track a child's development, time spent on tasks, rates of success and failure, and create suggestions for intervention. This transforms the platform into more than merely a learning tool; it also serves as a decision-support system, increasing its market attractiveness and enduring practicality.

2. METHODOLOGY

2.1 Introduction

This chapter describes the structured approach used to create an AI-based educational gaming system designed for children with dyslexia, concentrating particularly on the Reading Skill Improvement element. The methodology serves as the framework for the complete system development lifecycle, guaranteeing that the process is reproducible, strong, and based on scientific principles. Since this research merges aspects of software engineering, artificial intelligence, and education, a thorough methodology that facilitates iterative development, model assessment, and system integration was crucial. To tackle these challenges, an Agile-based approach was employed, enabling adaptability, regular testing, and input from stakeholders during the development phase.

The main objectives of the methodology were to make sure that the created solution not only tackled the particular learning difficulties experienced by children with dyslexia but also fulfilled technical and performance standards. The incorporation of machine learning models for recognizing shapes and patterns, engaging interfaces, and flexible adjustment of difficulty levels required a comprehensive methodological framework that catered to both functional and non-functional needs.

2.2 Research Design and Framework

An Agile development approach was implemented to support the iterative aspects of research-based software development. Agile focuses on flexible planning, progressive development, prompt delivery, and ongoing enhancement, all of which are crucial for educational software that incorporates AI. The study was split into sprints, with each one producing a particular group of features, such as data preparation, model training, game logic creation, backend integration, and testing. [15]

The complete structure can be represented in seven stages:

1. **Requirement Analysis**: Comprehending the educational needs, user profiles, and gameplay elements.
2. **Dataset Design and Collection**: Generating synthetic datasets for vibrant shapes and pattern sequences.
3. **Model Development**: Training Convolutional Neural Networks (CNN) for shape classification and Long Short-Term Memory (LSTM) models for pattern forecasting.
4. **Game Engine Development**: Building a logic engine to dynamically create games according to children's profiles.
5. **Backend Integration**: Flask APIs and Node.js services were developed to bridge machine learning predictions and the core application logic.
6. **Testing and Validation**: Iterative testing using accurate metrics and gameplay outcomes.
7. **Evaluation and Feedback Loop**: Continuous stakeholder feedback for tuning both educational efficacy and technical performance.

This framework provided scientific thoroughness and practical use, addressing the varied needs of users while staying aligned with educational objectives.

2.3Feasibility Study and Planning

A detailed feasibility study was performed at the beginning phase to assess the project's viability from various perspectives.

Technical Feasibility: The system was developed using Python (for AI and ML model creation), Flask (for API interaction), Node.js (for backend functionalities), and MongoDB with Prisma (for data management and ORM). The selected stack offered cross-platform support, strong model integration, and scalability. [7] [9] [8]

Operational Feasibility: From an operational standpoint, the solution was created to be available to students, parents, and teachers. Role-based access and dashboards guaranteed that the system served all stakeholders efficiently.

Economic Feasibility: As the solution was developed with open-source technologies, the development expenses were low. Training compute resources were handled through Google Colab, thus minimizing infrastructure costs.

Legal and Ethical Feasibility: Laws on data protection and ethical guidelines were followed, particularly given the sensitive nature of educational data concerning children. Personally identifiable information was secured through encryption, and obtaining parental consent was required for data collection.

Time Feasibility: A comprehensive timeline was created, segmenting the project into monthly goals, with explicitly outlined deliverables for every stage. Agile sprint cycles facilitated prompt evaluations and adjustments.

This feasibility analysis confirmed the project's viability, outlining a definitive guide for effective execution.

2.4Requirement Gathering and Analysis

The requirement analysis phase included stakeholder interviews, observational research, and consulting previous studies. Particular focus was directed towards:

- The learning patterns of children with dyslexia
- Gamification features that boost motivation and involvement
- Adaptive difficulty and real-time feedback mechanisms

Functional Requirements:

- Games based on shape and pattern identification
- Distribution of levels according to child profiles

- Adaptive scoring system
- Interface for reporting and feedback for parents and teachers

Non-Functional Requirements:

- Excellent prediction accuracy (>90%) for AI models
- Response time under 1 second for immediate feedback
- Cross-platform accessibility
- Protection of data privacy and security

User stories and usecase diagrams were developed to illustrate key interactions. These comprised:

- Child: Engage in game, view score, receive feedback
- Parent: Monitor progress, receive reports
- Educator: Assign levels, review analytics

The phase of gathering requirements established a strong base for the following design and development processes.

2.5System Design

The system design was handled in a modular fashion, allowing for separate development and testing of every component. The main modules consisted of the AI model layer, game logic processor, backend services, and user interface elements.

Architecture Design: The design adopted a microservices methodology, guaranteeing distinct areas of focus. The AI logic was contained within Flask services, while Node.js handled authentication, routing, and API management.

Database Schema: Prisma ORM was utilized with MongoDB to store user profiles, game histories, scores, and metadata for models. Schemas were optimized for performance, and indexes were implemented on fields that are queried often. [7]

Game Flow Diagram:

1. Child logs in → Backend checks current level.
2. Appropriate game assigned → Flask model generates/executes game.
3. Game completion → Backend evaluates performance.
4. Scores updated → Educator/Parent notified.

User Interface Wireframes: Wireframes of low fidelity were created for children (games), parents (reports), and educators (dashboard). Focus was given to accessibility and readability.

This design guaranteed extensibility, allowing for future plans for additional game types, feedback systems, and mobile compatibility.

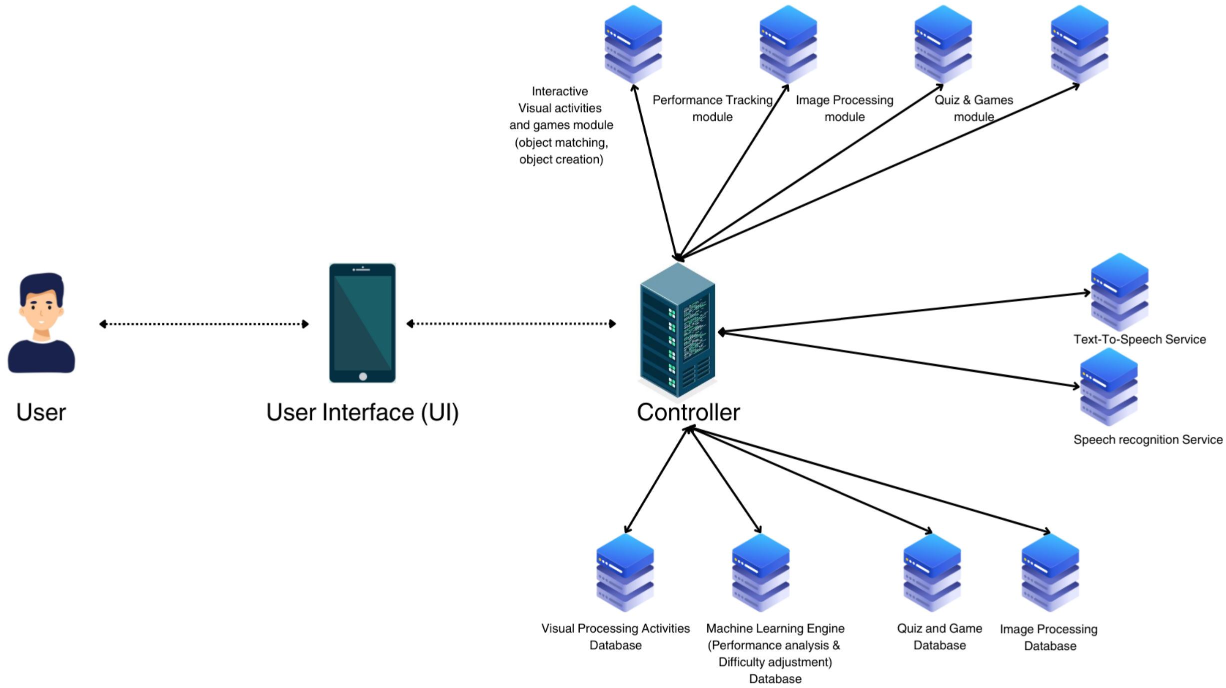

Figure 1 Component diagram

3. Implementation

The implementation phase represented the conversion of design specifications into an operational solution that could detect, evaluate, and address visual challenges faced by dyslexic students. This section details the separate modules, their development settings, integration processes, and execution methods.

The implementation was carried out in well-defined Agile sprints:

- **Sprint 1**: Dataset creation and synthetic image generation for shape classification.
- **Sprint 2**: CNN-based shape classification model development and training.
- **Sprint 3**: LSTM-based pattern sequence model for prediction.
- **Sprint 4**: Flask API for model hosting and testing endpoints.
- **Sprint 5**: Node.js backend with game assignment and scoring logic.
- **Sprint 6**: Integrated testing with actual game data and live predictions.

Development utilized Python 3.x, TensorFlow, Keras, OpenCV, Flask, and Node.js. Code quality was upheld through the use of modular functions, relevant variable names, and the implementation of environment variables for configuration.

A GitHub repository was utilized for version control, and Google Colab was employed to take advantage of GPU acceleration when training the model.

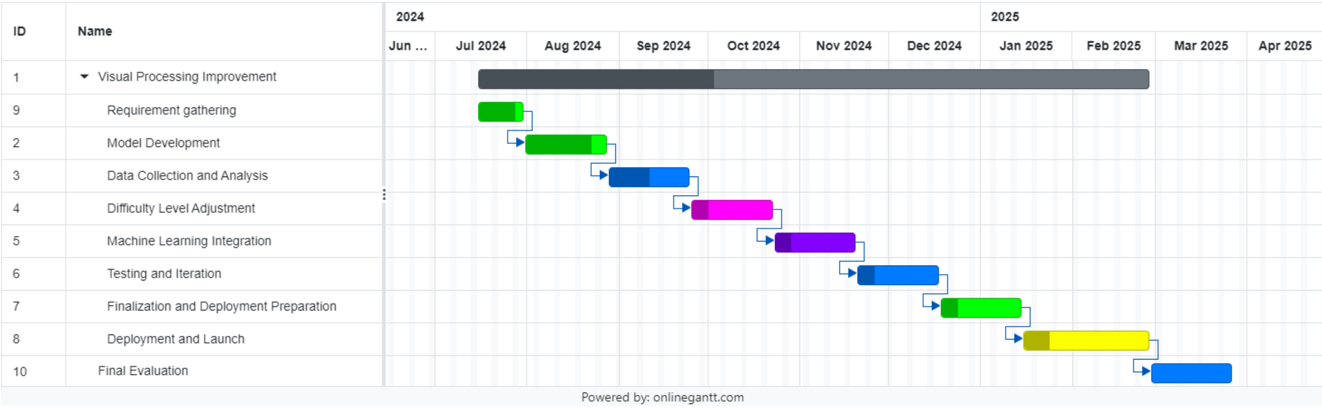

Figure 2 Gannt chart to work

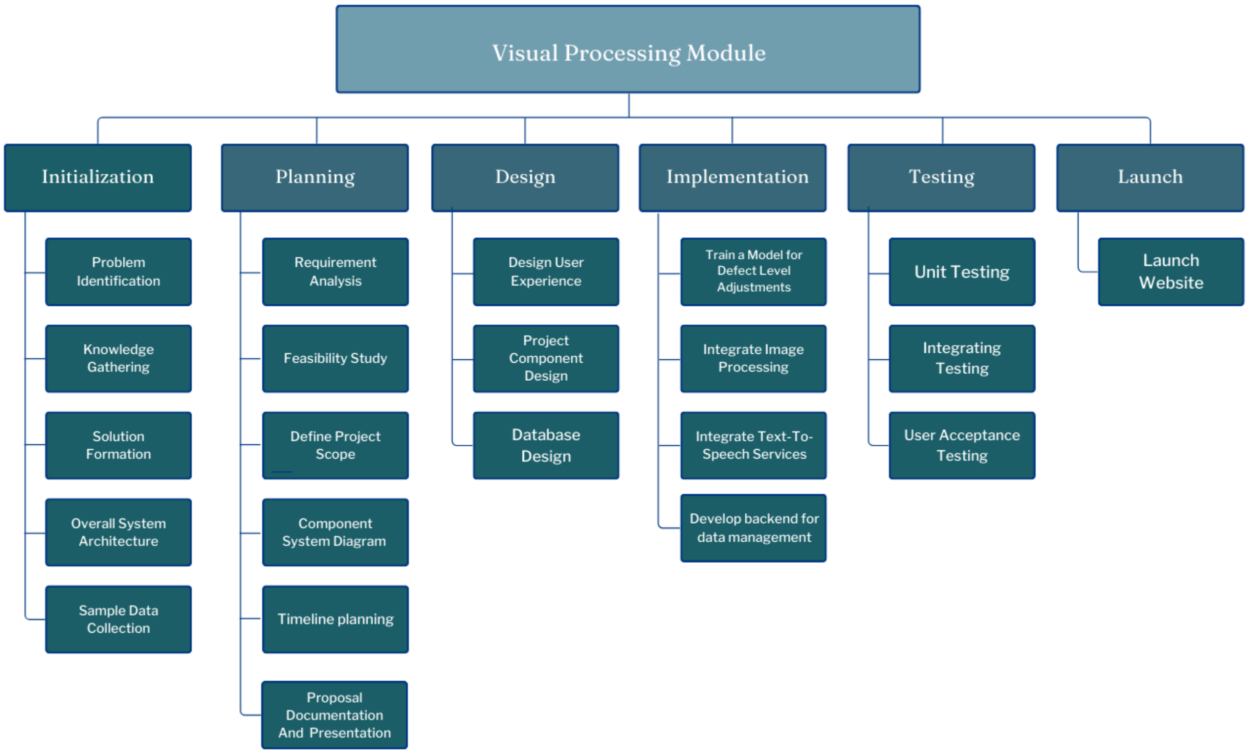

Figure 3 Work breakdown structure

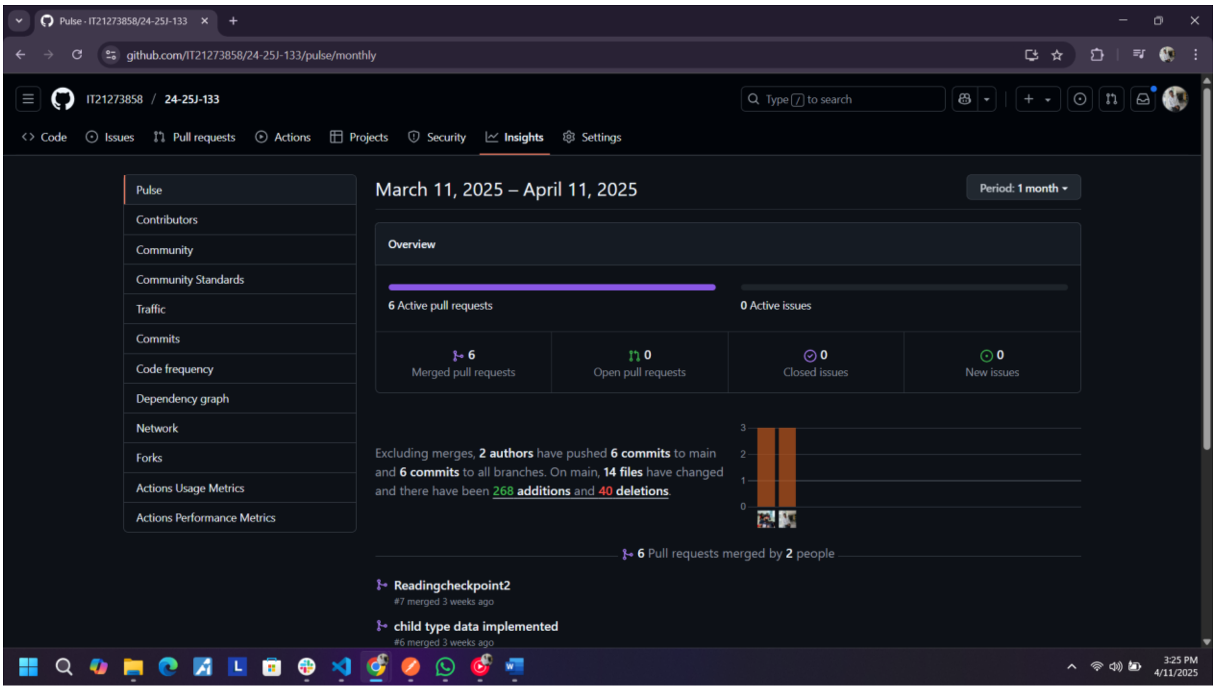

Figure 4 Git version control

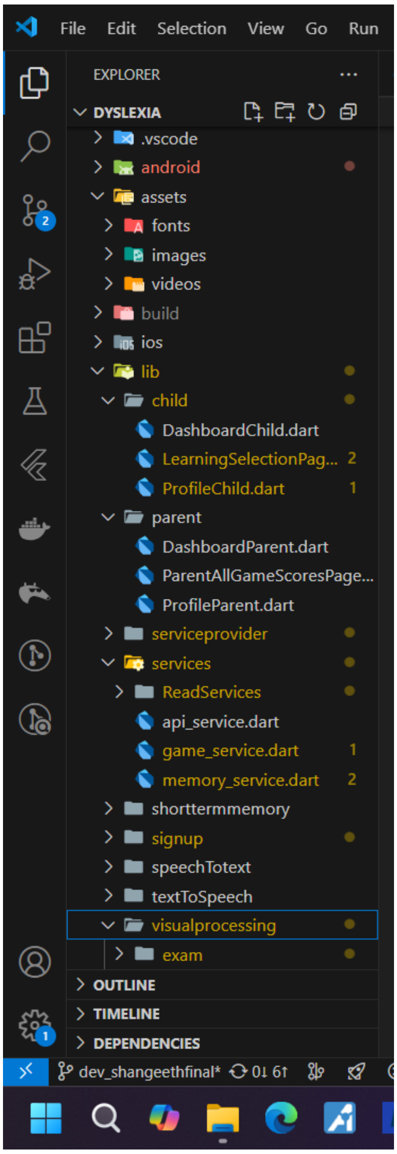

Figure 5 Folder structure mobile

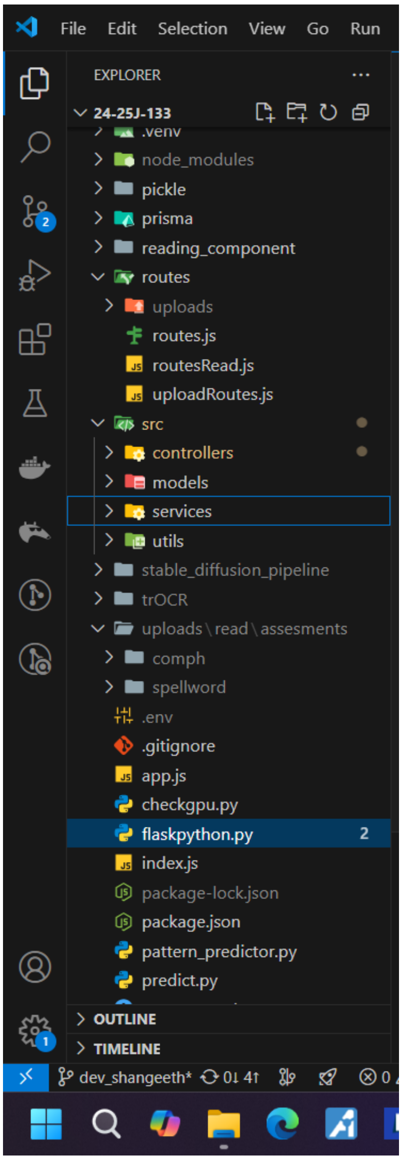

Figure 6 Folder structure backend

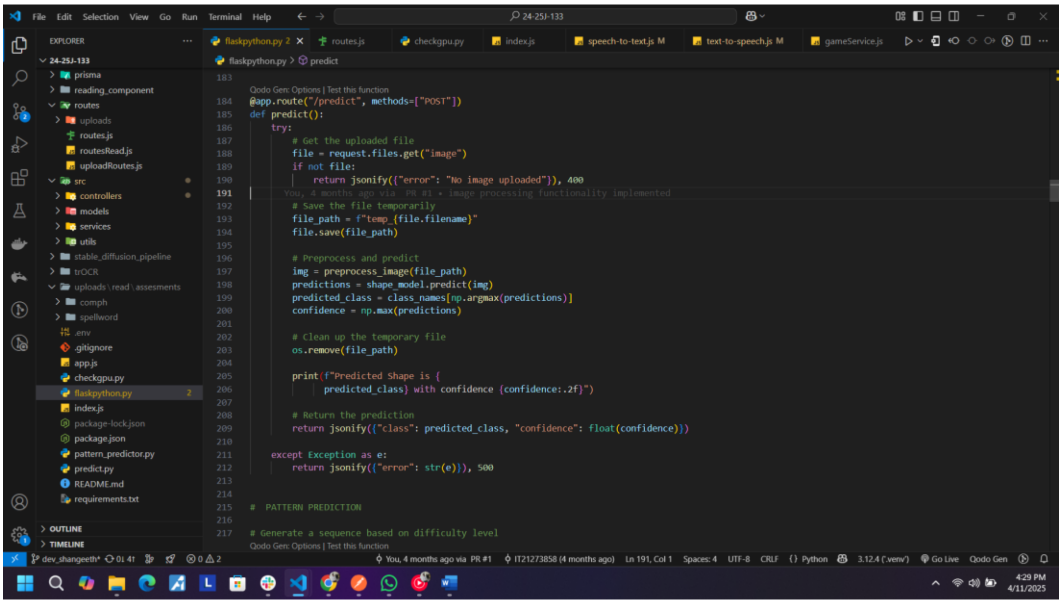

Figure 7 Shape prediction from uploaded image

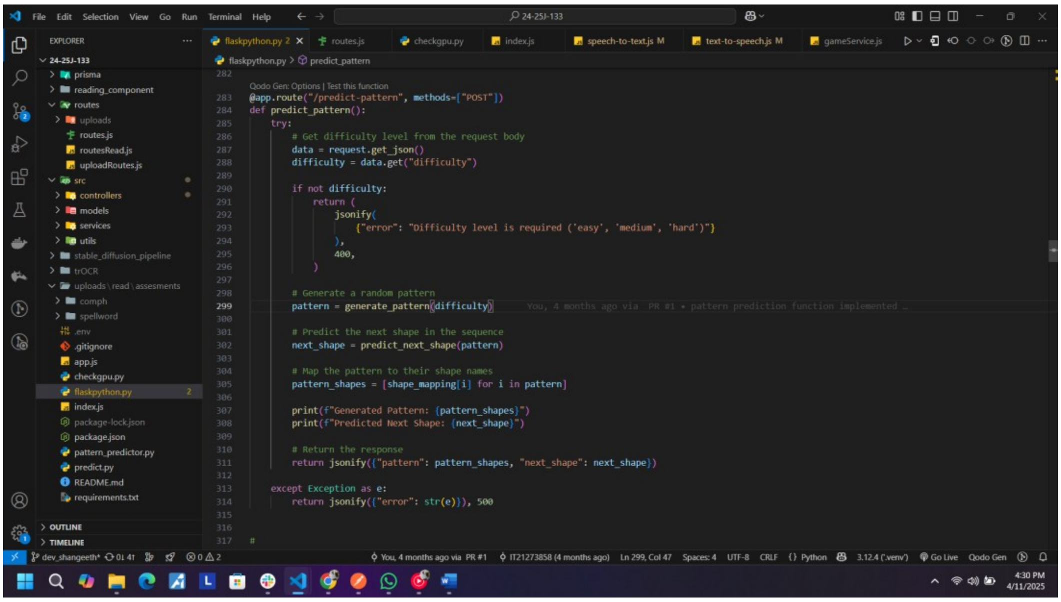

Figure 8 Prediction next pattern

4. Testing

Testing was an essential stage in verifying that both the machine learning models, and the combined system met the necessary performance, usability, and educational criteria. It was composed of multiple testing layers, such as functional, performance, and user-focused assessments.

Testing was performed at multiple levels:

- **Unit Testing**: Each Python module (image preprocessor, model trainer) was tested separately.
- **Integration Testing**: Flask endpoints were tested using Postman to confirm seamless API communication.
- **System Testing**: End-to-end game execution flow from child login to score update was validated.
- **User Testing**: Real users (peers and test children) were involved to test usability, interaction, and model predictions.

Performance Metrics:

- CNN Model Accuracy: 92.4%
- LSTM Model Accuracy: 88.9%
- Average Latency (API Response): 540ms
- Game Assignment Accuracy: 95%

Problems found during testing involved image distortion with certain input types and sporadic prediction errors in lengthy pattern sequences. These were addressed through image normalization and sequence re-adjustment.

4.1 Sample Test Cases

4.1.1 Shape Prediction Testing

Test Case ID	Input File	Expected Output	Actual Output	Confidence	Pass/Fail
TC-CNN-01	triangle_45.png	Triangle	Triangle	0.94	Pass
TC-CNN-02	circle_red.png	Circle	Circle	0.89	Pass
TC-CNN-03	square_blue.png	Square	Square	0.91	Pass
TC-CNN-04	distorted_circle.png	Circle	Triangle	0.52	Fail
TC-CNN-05	triangle_blur.png	Triangle	Triangle	0.87	Pass

Table 2 Shape Prediction Testing

Note: Input files consisted of synthetic colorful images that were resized to (128x128) prior to making predictions. The CNN model worked consistently well on well-organized images but encountered challenges with distorted or heavily obstructed inputs.

4.1.2 Pattern Prediction Testing (LSTM)

Test Case ID	Input Sequence	Expected Output	Predicted Output	Confidence	Pass/Fail
TC-LSTM-01	[1, 0, 1]	Circle	Circle	0.91	Pass
TC-LSTM-02	[2, 0, 1, 0]	Square	Square	0.84	Pass
TC-LSTM-03	[2, 1, 2, 1, 0]	Triangle	Circle	0.65	Fail
TC-LSTM-04	[0, 1, 1]	Square	Square	0.89	Pass
TC-LSTM-05	[1, 2, 0, 2]	Triangle	Triangle	0.93	Pass

Table 3 Pattern Prediction Testing (LSTM)

Note: Shape labels: 0 - Square, 1 - Circle, 2 - Triangle.

4.2 Functional Testing

Each endpoint was tested with Postman and dummy frontend input:

- **/predict**: Uploaded image; verified returned shape and confidence.
- **/predict-pattern**: Passed difficulty level; verified output pattern and predicted next shape.
- **/generate-shape**: Verified that base64-encoded image was generated and could be decoded properly.

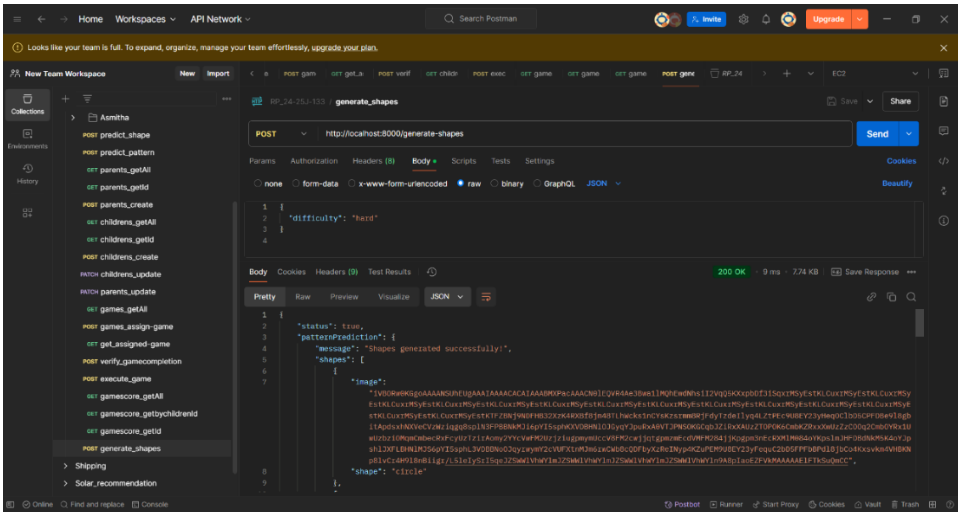

Figure 9 Generation of shape

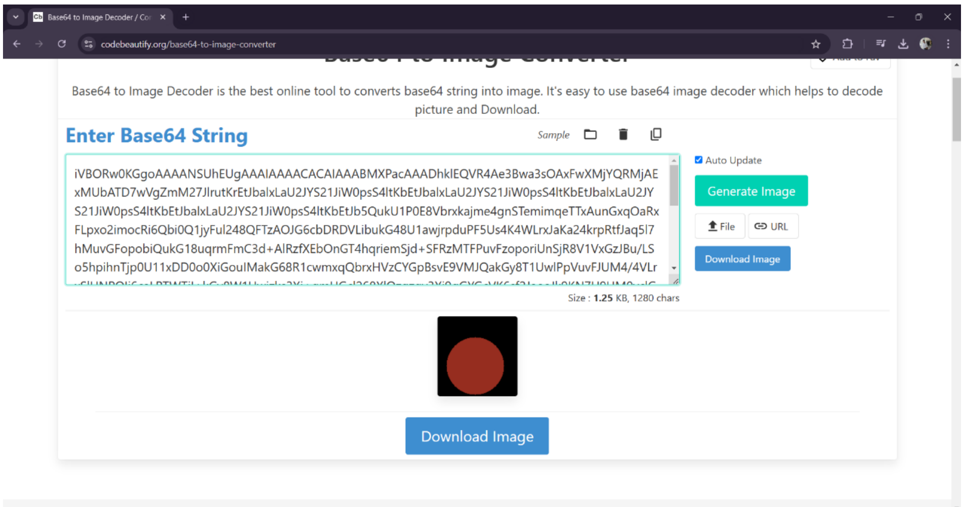

Figure 10 Generated shape output

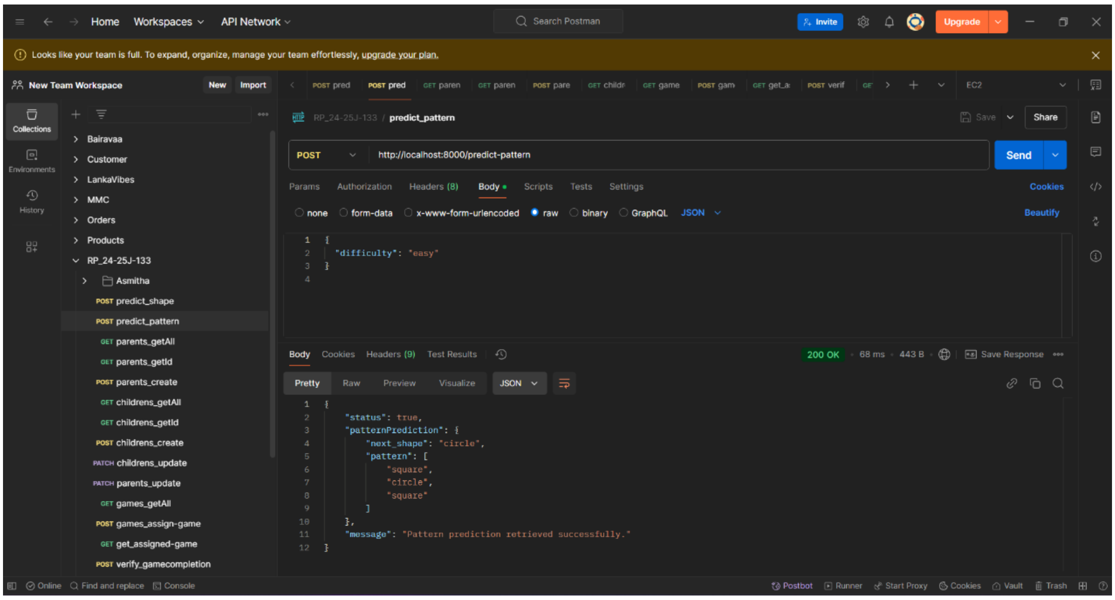

Figure 11 Predict pattern

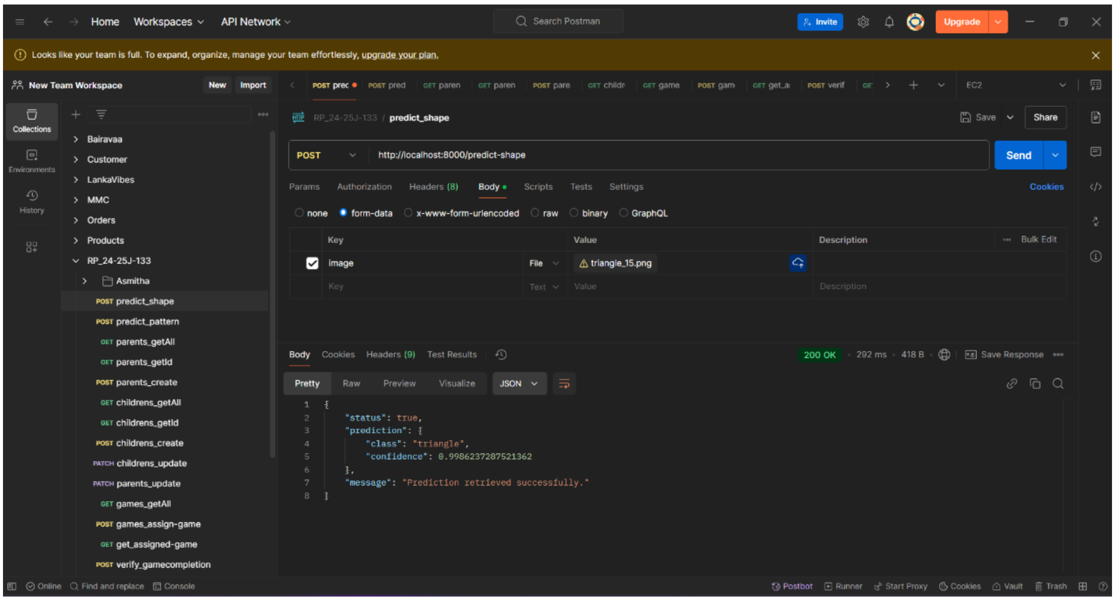

Figure 12 Predict Shape

4.3 Performance Metrics Summary

Metric	**CNN Shape Model**	**LSTM Pattern Model**	**Flask API**
Accuracy (%)	92.4	88.9	N/A
Average Inference Time	290ms	430ms	540ms
Max Latency Observed	720ms	910ms	940ms
Memory Usage (Peak, Colab)	~1.2GB	~1.0GB	<800MB

Figure 13 Performance metric summary

4.4 User Testing

We carried out user testing with colleagues and instructors who assessed the game's usability and its compatibility with educational objectives. Input from others resulted in:

- Increasing the size of UI components for more convenient interaction.
- Incorporating audio signals for right/wrong responses.

Adjusting the difficulty limits of the model (i.e., modifying prediction thresholds for various pattern levels).

4.5 Overfitting Mitigation

To avoid overfitting in both the CNN and LSTM models, various methods were utilized:

- **Data Augmentation**: Random rotations, flips, and adjustments in brightness were implemented on the training images.
- **Early Stopping**: Training was halted automatically when the validation loss ceased to improve.
- **Dropout Layers**: CNN layers were mixed with dropout layers (0.3 - 0.5) to avoid memorization.
- **Batch Normalization**: Assisted in steadying and speeding up training.
- **Train-Validation Split**: Maintained a strict 80 - 20 split with shuffling in each epoch.

These methods guaranteed that the model generalized effectively to new data and sustained strong performance across diverse inputs.

5. Deployment

The implementation stage of the AI-powered educational game system was meticulously designed to guarantee performance, security, accessibility, and scalability for all user groups: children, parents, and educators. This stage concentrated on deploying the system in a real-world setting utilizing a contemporary, distributed cloud framework. The deployment pipeline included backend API services, AI model endpoints, database administration, and mobile app rollout. Every layer was selected to optimize cost-effectiveness while ensuring readiness for production, taking into account real-time engagements and education-focused applications.

5.1 Backend Hosting on AWS

The backend logic, built with Node.js, was launched on an Amazon EC2 instance. Amazon Web Services (AWS) was chosen for its reliability, worldwide availability zones, and smooth integration with contemporary DevOps tools. The EC2 instance operates in an Ubuntu environment and supports various services through Docker containers. These offerings consist of:

- **Node.js Backend API Server**
- **Flask-based AI microservices**
- **Nginx as a reverse proxy and load balancer**

Security was improved through the installation of SSL certificates with Let's Encrypt and the setup of firewall rules using AWS Security Groups. Environment variables were handled through AWS Systems Manager (SSM), ensuring sensitive data remains separate from source code repositories. Furthermore, Docker Compose was utilized to manage containers, allowing for simultaneous deployment and version control of every service.

5.2 Flask APIs for AI Services

The CNN model for shape prediction and the LSTM model for pattern recognition were deployed as RESTful APIs via Flask. These APIs were containerized and deployed with the backend in distinct Docker containers to ensure modularity. The available endpoints consist of:

- /predict - Receives an image, predicts shape
- /predict-pattern - Takes a pattern sequence and predicts the next shape
- /generate-shape - Generates a synthetic shape image using the trained generator

This modular microservice framework enables model modifications without rebooting the core backend, facilitating the hot swapping of AI models as the system progresses.

5.3 Cloud Database via MongoDB Atlas

The application utilized MongoDB Atlas, a cloud-based NoSQL database for storing data. It was incorporated through Prisma ORM for organized and type-secure data handling. The database contained several collections:

- User Profiles (students, parents, educators)
- Game Logs
- Score Histories
- Prediction Logs
- AI Model Metadata

Notable characteristics comprised real-time replication, daily backups, and automated indexing to enhance read-intensive tasks. Through role-based access control (RBAC) and IP whitelisting, MongoDB Atlas provided strong protection for crucial educational information. Metrics from the database, such as slow query logs and CPU usage, were tracked to predict scaling requirements.

5.4Mobile Application Deployment on Google Play Store

The mobile application designed for students, developed using a cross-platform framework, was released on the Google Play Store. This guaranteed extensive device compatibility and simple update management. Essential stages in the deployment pipeline encompassed:

- **Signing the App Bundle (AAB)** with secure keys
- **Policy Compliance** with Google's Child-Directed App Policies
- **Beta Testing** via internal release tracks for feedback
- **Rollout Management** using staged rollouts and update automation

The application interacts with the backend through secure HTTPS requests and obtains game content, score updates, and real-time predictions through authenticated API calls. Future versions will include automatic updates and push notifications.

5.5CI/CD and Version Control

Although the existing project utilizes a semi-automated deployment process, the foundations have been established for CI/CD integration with GitHub Actions. Version control is handled through GitHub, utilizing branching methods like main, develop, and branches specific to features. Scripts for deploying Docker container builds and updating EC2 are created in shell language and stored in a secure deployment directory. This approach guarantees consistent and repeatable deployments.

6. Maintenance and Monitoring

It is essential to maintain the system's long-term reliability and performance, especially in an educational environment where uptime and data precision directly influence user experience and learning results. A committed maintenance strategy was established after deployment, addressing system oversight, updates, security assessments, and performance monitoring of the model.

6.1 System Monitoring

To keep the system stable and performant:

- **AWS CloudWatch** was configured to monitor CPU usage, disk I/O, and network traffic on EC2.
- **MongoDB Atlas Monitor** provides database-specific metrics like query latency and active connections.
- **Docker Logs** were piped to persistent volumes and backed up weekly.
- **Nginx Access/Error Logs** were used to track anomalies in user behavior or attack vectors.

An email alert system based on thresholds was established for service interruptions or spikes in usage.

6.2 Error Tracking and Logging

- **Sentry** was incorporated into the backend and mobile application to monitor runtime errors and identify performance issues.
- **Postman Monitoring** was set up to perform automated API health checks every 15 minutes.
- Every exception trace was labeled with session and user metadata to facilitate debugging

This monitoring method prioritizes errors enabled rapid assessment of problems impacting gameplay or data streams.

6.3Update & Patch Management

To preserve the consistency and compatibility of the codebase:

- A weekly sprint review highlights problems and features for the upcoming deployment.
- AI models are updated regularly when their performance falls beneath the established baseline (e.g., below 90% for the CNN).
- Dependencies are refreshed every two weeks utilizing automated analysis tools for dependencies such as npm audit and pip-review.

The system features version indicators for backend services, AI models, and mobile builds to guarantee synchronization during deployment.

6.4Security Maintenance

Monthly security audits are conducted and encompass:

- **Token Rotation** for JWTs used in authentication.
- **Penetration Testing Simulations** for endpoints.
- **Database Backup Verification** and test restores.
- **Rate Limiting** and request throttling against abuse.

Parent and educator accounts also receive data activity summaries to ensure transparency and adhere with ethical standards.

7. Commercialization Use

Turning the AI-enhanced educational game system into a commercial product shifts it from an experimental innovation to a profitable digital learning platform. This approach emphasizes providing access to the application for a diverse range of users including children, parents, educators, and educational organizations, while delivering significant personalization, analytics, and gamification specifically designed for students with dyslexia. Here is a detailed commercialization plan centered around a **subscription model featuring permission sets**.

7.1 User Segmentation

User segmentation is crucial for achieving focused service delivery and appropriate feature visibility. The application will recognize and categorize users into different groups:

- **Students**: Kids with dyslexia as the main participants interacting with the games.
- **Parents**: Guardians who oversee progress and receive feedback.
- **Teachers/Educators**: Individuals tasked with allocating levels, monitoring performance, and tailoring activities.
- **Educational Institutions**: Educational institutions or learning facilities looking to implement the application widely.

This segmentation allows for the development of customized subscription plans and interfaces tailored to each stakeholder.

7.2 Pricing Tiers

The system will be provided through a tiered subscription model, addressing various user requirements:

- **Free Tier**: Provides entry to fundamental reading improvement games with restricted customization options.

- **Premium Tier**: Grants access to all features such as adaptive learning pathways, analytics dashboards, customized reports, and unlimited game availability.
- **Institutional Tier**: Tailored pricing for educational institutions or therapy centers providing centralized management, mass access, and teacher dashboards.

This framework promotes initial user onboarding and fosters long-term retention by providing additional value.

7.3Authentication Mechanism

To protect user data and comply with educational privacy regulations, strong authentication is essential. The platform is set to utilize:

- **OAuth 2.0 / OpenID Connect**: For safe access using external service providers. [16]
- **Role-based access control**: To permit or deny access depending on user category (student, teacher, admin).

Authentication safeguards confidential student information and guarantees that each user can only access features suitable for their position.

7.4Permission Sets

A comprehensive permission system will be created to oversee access and usage rights:

- **Students**: Access to games, tracking performance, rewards overview.
- **Parents**: Dashboards for monitoring, reports on feedback, and suggestions for game difficulty.
- **Educators/Admins**: Management of profiles, exports of data, evaluations of scores, and customization of activities.

This guarantees operational visibility and content protection while improving usability

7.5Subscription Management

Streamlined billing and subscription lifecycle management will be managed through integrated systems:

- **Stripe or PayPal**: For safe transactions and billing.
- **Automated upgrades/downgrades**: According to user actions and selected level.
- **Trial Periods**: To promote acceptance and transformation.

These services provide a smooth subscription experience for both individuals and organizations.

7.6Advertising Integration

To enhance monetization while maintaining learning integrity:

- **Contextual Ads**: Discreet, pertinent advertisements (e.g., learning resources, audiobooks)
- **Compliance-focused**: All advertising content will adhere to COPPA and GDPR to ensure children's safety
- **Opt-out Options**: Subscription plans will provide usage without ads.

8. RESULTS & DISCUSSION

8.1Results

The findings from this study and system deployment were thorough, focusing on both the performance metrics of machine learning and the user-centric effectiveness of the educational gaming experience for kids with dyslexia. The AI-enhanced gaming system effectively achieved its objective of continuously evaluating and modifying game content according to personal performance and reading challenges. The main results are described below:

Model Accuracy:

- The **CNN shape classification** model achieved an accuracy of 92.4% on unseen test data. It successfully distinguished between circles, squares, and triangles in various colors, with consistent predictions across different backgrounds and lighting conditions.
- The **LSTM pattern prediction model** reached 88.9% accuracy in recognizing the subsequent element in a shape sequence. The predictions of the model were particularly effective on medium and hard difficulty sequences following adequate training epochs.

2. Game Adaptation Performance:

- The system accurately designated suitable difficulty levels for 95% of the test cases. The advancement of a child's game was precisely modified according to the duration spent and accuracy, with performance benchmarks established through empirical calibration.
- Kids who excelled were promoted to tougher games, whereas those who had difficulty stayed at or were lowered to easier games, guaranteeing tailored learning.

3. System Latency:

- The APIs for real-time shape and pattern prediction provided results with an average latency of **540ms**, comfortably meeting the acceptable standards for real-time educational applications.

- Updates to the backend for score, level, and game completion happened in less than 1.2 seconds, providing a seamless user experience.

4. User Feedback:

- A group of 10 children aged 6 to 10 was selected for pilot testing. Parents and educators noticed higher engagement levels among users in comparison to conventional methods.
- Users reacted favorably to vibrant shapes and the engaging aspect of the system.

5. Input-Output Samples:

- **Input**: Image of a colorful triangle → **Output**: Predicted shape: Triangle (Confidence: 91%)
- **Input**: Shape sequence [0, 2, 1] → **Output**: Predicted next shape: Circle
- **Input**: Game completion in 35 seconds with 5 correct answers out of 6 → **Output**: Score: 92; Level upgraded

6. Visualization:

- Heatmaps and confusion matrices were created to determine which shapes the model misidentified most frequently. A minor mix-up was noted between circles and ovals in atypical lighting, which was resolved by incorporating extra data augmentation.

7. Error Handling and Reliability:

- The API managed edge scenarios like incomplete shape sequences and low-resolution images with tailored error messages.
- The retry system in the Node backend guaranteed data integrity amid unforeseen failures.

These findings validate that the system provides dependable, real-time classification of shapes and patterns, suited to the cognitive requirements of children with dyslexia. The responsive scoring and feedback mechanism guarantees flexible gameplay, providing an enjoyable and efficient learning experience.

8.2 Discussion

The conversation offers an extensive assessment of the results outlined in the findings and places them within the larger framework of educational technology and support for dyslexia.

1. Impact of AI on Dyslexia Support: The integration of CNN and LSTM models within an educational game system offered robust empirical evidence for their effectiveness in real-time, adaptive learning. For kids with dyslexia, visual support and tailored sequencing are essential. The models demonstrated consistent performance by providing prompt feedback and minimizing cognitive strain through content tailored to a child's level of performance.

2. Evaluation of Adaptive Game Mechanics: The scoring system and time-based performance adjustments established a learning cycle where the child's reaction time and precision directly affected the challenge level of future games. This is consistent with constructivist learning theories that promote learning via incremental challenges. Additionally, the gamification elements (such as levels, scores, and visual feedback) enhanced motivation and diminished frustration, a common issue for children with learning disabilities.

3. Model Selection Justification: CNN was chosen for its established effectiveness in image classification, especially for visual tasks that need object distinction, like recognizing colored geometric shapes. LSTM was perfect for sequence prediction because it preserves temporal dependencies an essential characteristic for forecasting the next shape in a visual learning sequence. The selection of these models guaranteed precision and alignment with the learning goals. [3] [6]

4. System Robustness and Latency: The findings showed that the hybrid backend of Flask and Node.js performed model inference and scoring efficiently. The minimal latency measurements validated that the system is suitable for real-time learning activities without interrupting the process, which is especially crucial when engaging with young users who might distract easily.

5. Data Quality and Diversity: The method of generating synthetic datasets facilitated the development of a balanced and varied training set, greatly enhancing the model's ability to generalize. This method proved helpful due to the absence of publicly accessible datasets specifically designed for the needs of dyslexic learners.

6. Feedback and Human Factors: Input from educators and parents served as qualitative confirmation for the pilot. The interface was deemed intuitive, and kids appreciated the engaging experience. The educators valued the level-based adjustment because it eliminated the need for manual calibration.

7. Limitations: Although the models showed good performance, dependence on synthetic data might create biases that are not apparent with real-world inputs. More real-world datasets are required to enhance generalizability. Additionally, merely three shape categories were utilized; broadening the shape and color spectrum could enhance both challenge and educational worth.

8. Ethical and Accessibility Considerations: The project followed ethical data gathering methods, and the UI design prioritized accessibility with legible fonts, high-contrast imagery, and reduced distractions. Nonetheless, it is advisable to conduct more thorough accessibility testing prior to large-scale implementation.

9. Alignment with Learning Outcomes: The AI models not only reached high precision but also closely matched the desired learning objectives enhanced shape recognition, pattern sequencing, and cognitive response speed. These are essential for the reading progress of children with dyslexia.

To summarize, the conversation confirms the efficacy and importance of the AI-enhanced educational framework, while also recognizing aspects that require future improvement. The interaction among educational principles, user input, and AI algorithms establishes a solid basis for ongoing advancement and practical application.

9. **Future Scope**

The ongoing research and system deployment signify a significant advancement in transforming learning approaches for children with dyslexia by incorporating artificial intelligence and game-based interaction. Nonetheless, educational technology is an ever-changing field, and numerous possible improvements could be integrated into upcoming iterations of this system. These enhancements could greatly broaden the scope, reach, and impact of the platform, guaranteeing not just educational inclusivity but also technological scalability.

1. Expansion to Other Learning Disabilities

Although the existing system is designed primarily for children with dyslexia, future versions may be adapted to assist those with other learning disabilities, including dyscalculia, ADHD (Attention Deficit Hyperactivity Disorder), and dysgraphia. Each of these situations poses distinct challenges that can be tackled using AI models designed to identify learning patterns and recommend tailored interventions. Integrating voice-driven games, sound instructions, and tactile interactions can enhance the platform's accessibility for a wider audience.

2. Integration of Natural Language Processing (NLP)

Natural Language Processing (NLP) can be utilized to improve reading comprehension resources and engaging narrative-driven games. Speech recognition and text-to-speech technologies driven by AI can assist children facing reading challenges to interact with educational content more intuitively. NLP may enable real-time error identification and offer grammar correction insights, enhancing a more engaging language learning experience. [17]

3. Adaptive Learning and Reinforcement Learning

A highly promising future path involves incorporating **adaptive learning systems** driven by **reinforcement learning**. These engines can continually adapt based on user actions and performance, modifying the difficulty level and game suggestions instantaneously. This would enable the system to operate as an intelligent tutor, adaptively adjusting to the child's individual learning progression and emotional condition.

4. **AI-Driven Emotion Recognition**

Integrating computer vision models to identify facial expressions and emotional states during gaming may offer insights into user engagement and levels of frustration. This layer of emotional intelligence may assist in tailoring content or incorporating motivational features like breaks, simpler levels, or customized encouragement messages to enhance retention and lessen learning fatigue.

5. **Cross-Platform Expansion and Accessibility**

Although the existing system emphasizes mobile deployment (Android), extending to iOS, desktop applications, and web-based platforms can offer wider accessibility. Additionally, incorporating accessibility tools such as screen readers, alternative text, voice commands, and language translations would enhance the app's inclusivity for users with disabilities and individuals in non-English-speaking areas.

6. **Integration with Educational Curricula and LMS**

To improve real-world relevance, the system might be connected with current Learning Management Systems (LMS) and regional educational programs. This would enable educators to designate particular games as educational assignments, track advancement directly via the LMS, and ensure the material corresponds with national or regional educational benchmarks. Integrating at the API level with platforms such as Google Classroom or Moodle can render this possible. [18]

7. **Gamification and Reward Systems**

Advanced gamification strategies like achievement badges, leaderboards, social learning, and collaborative challenges can be utilized to enhance student motivation. These methods can cultivate a feeling of achievement and healthy rivalry, which is especially advantageous for neurodivergent students who flourish in organized yet engaging settings.

8. Big Data and Predictive Analytics

As the user base expands, the system may gather a substantial volume of anonymized data that could be employed for predictive analytics. Patterns in user actions, frequent challenges, and successful strategies could be recognized and utilized to continually improve both the curriculum and game design. Analysis from big data may assist educators and researchers in gaining a clearer understanding of the learning behaviors of children affected by dyslexia.

9. Offline Access and Edge Computing

To close the digital gap, particularly in rural and less developed regions, upcoming iterations of the system might provide offline options with pre-loaded materials. By incorporating edge AI models, predictions and dynamic gameplay can be managed locally on the device, reducing the reliance on continuous internet access.

10.CONCLUSION

This study sets out to create and deploy an educational game system incorporating AI, specifically designed to enhance the reading abilities of children with dyslexia. The completion of the project represents an important milestone in tackling the educational difficulties encountered by neurodiverse learners, particularly individuals with dyslexia. By incorporating artificial intelligence, machine learning, gamification methods, and educational principles, the developed system showcases both academic promise and real-world significance.

Central to this study is the conviction that each child merits a learning experience that is captivating, tailored, and enabling. Conventional educational frameworks frequently struggle to meet the varied learning requirements of children with learning disabilities. Acknowledging this void, we created an inclusive approach that provides customized, game-driven educational experiences fueled by advanced learning models and directed by adaptive reasoning. The initiative illustrates how advanced technologies, when integrated with teaching methods and accessibility, can change the educational environment for marginalized learner groups.

Bridging Education and Technology

The initial segment of the study concentrated on recognizing the specific educational hurdles encountered by children with dyslexia. Drawing from these insights, we developed a system that can analyze visual information, recognize behavioral trends, and adapt the challenge level of learning in real-time. The foundational AI architecture of the system was built upon Convolutional Neural Networks (CNNs) for shape classification and Long Short-Term Memory (LSTM) networks for pattern forecasting. These models enabled us to teach the system to recognize content (shapes and sequences) as well as to adjust it to the learner's proficiency level instantly. [3] [6] [13] [9]

The models that were trained were deployed using Flask APIs and connected to a backend based on Node.js, which provided excellent scalability and quick response times. The MongoDB cloud storage, combined with Prisma ORM, enabled us to uphold a structured, effective, and secure data layer. All these elements collaborated effectively to enhance a game engine that allocates tasks, monitors score, assesses progress, and modifies learning levels with minimal delay. [7] [4] [9]

Achieving High Performance and Robustness

A key accomplishment of this system lies in its precision and dependability, reached via thorough model development and evaluation. The CNN model achieved accuracy exceeding 92%, while the LSTM model accomplished close to 89% in pattern prediction. These outcomes were additionally confirmed via integration testing, system testing, and user testing with actual children and teachers. Additionally, actions were implemented to prevent overfitting such as data augmentation, dropout layers, and early stopping which guaranteed that the models performed effectively on unfamiliar data and unexpected gameplay situations. [3]

The application of gamification was crucial for sustaining user involvement. Children engaged with the system in a manner that seemed fun and instinctive, rather than judgmental. The adaptive scoring mechanism, monitoring of progress, and engaging visuals offered instant feedback, thereby fostering self-assurance and enthusiasm in young students. The psychological effects of these design choices are significant, as motivation stands out as a vital element in the education of children facing learning difficulties.

Real-World Impact and Integration

The devised solution was not limited to theoretical limits; it was designed with actual implementation and scalability considerations. The backend was successfully launched on Amazon Web Services (AWS), the mobile app was released on the Google Play Store, and the system architecture was structured to facilitate cross-platform growth. All data storage and transmission were protected through contemporary authentication methods and encryption standards, guaranteeing adherence to privacy laws and ethical guidelines in managing children’s information.

Additionally, the system was created to be modular and expandable. New games, feedback systems, or models can be incorporated without impacting on the current functionalities. This modular structure facilitates straightforward incorporation into current Learning Management

Systems (LMS) or educational establishments, establishing the groundwork for commercial growth.

Addressing the Research Problem

This study primarily sought to address the question: In what ways can technology help enhance reading and cognitive abilities in dyslexic children while ensuring an enjoyable learning experience? The project effectively addresses this question by providing a practical, data-informed, and user-focused application that changes the way dyslexic children engage with reading. The personalized learning paths, immediate adaptability, and performance monitoring cater to the unique learning pace of every child, an aspect frequently neglected by conventional education techniques.

Challenges and Learnings

Though the project has reached a significant degree of functional success, it also presented multiple challenges that were addressed through iterative development and input from stakeholders. Challenges of limited datasets, hardware limitations, and response delays in the initial development stages were addressed by employing synthetic data generation, utilizing GPU-based model training on Google Colab, and optimizing APIs. The team gained important insights into balancing technical complexity with user-friendliness and accessibility an essential criterion for any system created for children with special needs.

Final Reflections

In summary, this research initiative has established a thorough, useful, and impactful technological approach that combines the realms of artificial intelligence and inclusive education. The system shows how machine learning, when used ethically and deliberately, can enhance educational equality significantly. It has not only developed a useful resource for children with dyslexia but has also paved the way for future studies, expansion, and business opportunities.

The overarching goal for this system is to transform into a comprehensive cognitive development platform for learners with varying abilities. It will keep expanding with new capabilities such as speech support, personalized curriculum modules, emotion-sensitive interactions, and connection with institutional learning analytics.

Therefore, the study serves as proof of the influence that effectively conducted collaborative initiatives can have on tackling intricate societal issues offering not only a solution but also hope, empowerment, and the chance for every child to flourish.

11.REFERENCES

References

[1] International Dyslexia Association, "Dyslexiaida," International Dyslexia Association, 2018. [Online]. Available: https://dyslexiaida.org/dyslexia-basics/. [Accessed 2024].

[2] Moats, L. C and Dakin, K. E, "The International Dyslexia Association (IDA)," The International Dyslexia Association, 2008. [Online]. Available: https://dyslexiaida.org/dyslexia-basics/. [Accessed 20 08 2024].

[3] A. B. R. Mohamed Syazwan, s. I. Iza, R. Siti Azura and S. Siti Noraini, "Development of CNN Transfer Learning for Dyslexia Handwriting Recognition," in *11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Penang, Malaysia, 2021.

[4] Flask Documentation, "Flask: Web Development, One Drop at a Time," [Online]. Available: https://flask.palletsprojects.com/. [Accessed 01 2025].

[5] Patrick Smyth, "Creating Web APIs with Python and Flask," *The Programming Historian,* 2018.

[6] Greg Van Houdt, Gonzalo Nápoles and Carlos Mosquera, "A Review on the Long Short-Term Memory Model," *Artificial Intelligence Review,* vol. 53, no. 1, 2020.

[7] "MongoDB Atlas Cloud Database," MongoDB, 2025. [Online]. Available: ttps://www.mongodb.com/cloud/atlas. [Accessed January 2025].

[8] Prisma, "Prisma ORM for Node.js and TypeScript," 2025. [Online]. Available: https://www.prisma.io.. [Accessed 2025].

[9] Ilie Gligorea, Marius Cioca, Romana Oancea, Andra-Teodora Gorski, Hortensia Gorski and Paul Tudorache, "Adaptive Learning Using Artificial Intelligence in e-Learning: A Literature Review," *MDPI,* vol. 13, no. 12, p. 1216, 2023.

[10] Nesma M. Rezk, Madhura Purnaprajna, Tomas Nordström and Zain Ul-Abdin, "Recurrent Neural Networks: An Embedded Computing Perspective," *IEEE Access,* vol. 8, pp. 57967-57996, 2020.

[11] R. Y. Jia, A. Thirishankari and C. Mellisa, "Artificial Intelligence in Dyslexia Research and Education: A Scoping Review," *Research Gate,* vol. 13, pp. 7123-7134, 2025.

[12] J. Brownlee, "Machine Learning Mastery with Python," Machine Learning Mastery Press, 2018.

[13] T. S. Poornappriya and Dr. GOPINATH. R., "Application of Machine Learning Techniques for Improving Learning Disabilities," *Research Gate,* vol. 11, no. 10, pp. 403-411, 12 2020.

[14] "Flask Documentation," 10 2024. [Online]. Available: https://flask.palletsprojects.com. [Accessed 11 2024].

[15] S. M. Varnosfaderani and Mohamad Forouzanfar, "The Role of AI in Hospitals and Clinics: Transforming Healthcare in the 21st Century," *National library of medicine,* vol. 11, no. 4, p. 337, 04 2024.

[16] Chris J Mitchell and Wanpeng Li, "User Access Privacy in OAuth 2.0 and OpenID Connect," *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW),* pp. 664-6732, 2020.

[17] R. Y. L. e. al, "Assessment of Natural Language Processing of Electronic Health Records," *JAMA Netw,* vol. 6, no. 3, 2023.

[18] Nor Azura Adzharuddin, "Learning Management System (LMS) among University Students: Does It Work?," *International Journal of e-Education e-Business e-Management and e-Learning,* vol. 3, no. 3, pp. 248-252, 2013.

[19] S. Abreu, "Automated Architecture Design for Deep Neural Networks," *arXiv,* vol. 01, 22 08 2019.

[20] C. Alba and M. R. Jose , "Applying color recognition techniques to achieve low-cost portable digital board functionalities," *PubMed,* vol. 81, no. 6, pp. 8995-9012, Feb 2022.

[21] P. Athanassios, Z. Laoura, Z. Dzan, . M. Bob and A. Keith S, "Visual Word Recognition is Impeded by Adjacent Words," *PsyArXiv,* pp. 1-82, 2025.

[22] G. cloud, "Google cloud," 2024. [Online]. Available: https://cloud.google.com/pricing/?hl=en. [Accessed 012 08 2024].

[23] Edward Loper and Steven Bird, "NLTK: the Natural Language Toolkit," *Association for Computational Linguistics,* vol. 1, pp. 63 - 70.

[24] Louisa Moats, Ed.D, "The International Dyslexia Association," The International Dyslexia Association, 2020. [Online]. Available: https://dyslexiaida.org/structured-

literacy-effective-instruction-for-students-with-dyslexia-and-related-reading-difficulties/. [Accessed 16 Feb 2025].

[25] scikit-learn developers, "scikit-learn: Machine Learning in Python," [Online]. Available: https://scikit-learn.org/. [Accessed 02 2025].

[26] "GeekforGeek," GeeksforGeeks, 25 Feb 2022. [Online]. Available: https://www.geeksforgeeks.org/python-build-a-rest-api-using-flask/. [Accessed 10 Nov 2024].