



System Overview and Architecture

GameOn: Gamified Reading Learning is a mobile application developed to enhance the reading skills of Grade 2 students through interactive, gamified learning activities. The app provides students with reading-based games and exercises that reward them with points and achievements, fostering both engagement and learning progression.

One of the core innovations in the system is the integration of a machine learning prediction model using TensorFlow.js, which forecasts a student's performance based on their recent reading scores. This predictive feedback helps students visualize their learning trend and motivates them to keep improving.

The system also tracks student performance over time, and presents their results through graphs and progress reports, making it easier for teachers and parents to monitor reading development.

GameOn uses a **modular and layered architecture** combining mobile development, cloud services, and machine learning:

1. Frontend (React Native + Expo)

- a) Built with React Native and tested on Expo Go, the frontend presents a user-friendly interface for young learners.
- b) Handles navigation, user authentication, game mechanics, and visualization of scores and predictions.

2. Backend (Firebase)

- a) Authentication: Firebase Auth manages secure user login for students.
- b) Database: Firestore stores user profiles, reading scores, and daily activity logs in real time.
- c) Storage: Could be used for storing images, avatars, or other user-generated content if needed.

3. Machine Learning Module (TensorFlow.js)

- a) A linear regression model is built and trained using TensorFlow.js directly within the mobile app.
- b) It uses the student's recent scores (last 2 days + today) to predict performance for the next 5 days.
- c) This model retrains every time the student accesses the "Statistics" feature.

4. Data Visualization

- a) Utilizes react-native-chart-kit to display past and predicted scores in a clear, child-friendly line graph.
- b) Shows trends in performance to keep learners and teachers informed.



Component	Description
Input	Daily reading activities and the points scored by the user
Process	1) Score tracking 2) Graph rendering 3) Machine learning prediction (TensorFlow) 4) Pre/Post tests and evaluation
Output	1) Student's current and predicted performance 2) Graphs 3) Insights into reading behavior and improvement

Summary of Enhancements and Rationale

The main enhancement implemented in the GameOn application is the integration of a machine learning-powered score prediction system using TensorFlow.js. This new feature empowers the app to forecast a student's expected performance over the next five days, based on their most recent reading activity scores stored in Firebase.

This enhancement includes:

- A linear regression model that trains on each student's latest three scores (2 past days and today).
- Real-time model training and prediction triggered by the user's action (pressing the "Statistics" button).
- A graphical visualization of both recent and future performance using an interactive line chart.
- Code optimization to improve model performance and reduce app latency during predictions.

The rationale for introducing this enhancement stems from the need to:

1. Motivate Continuous Learning

By predicting future scores, students receive feedback that encourages consistency and effort. It gamified improvement by showing potential progress.

2. Promote Data-Driven Learning Insights

The feature helps students, teachers, and parents understand reading progress more clearly through visual trends instead of just raw scores.

3. Personalize the Learning Experience

Predictions are personalized based on each user's historical data, making the system adaptive rather than generic.

4. Introduce AI Concepts in Education

Integrating machine learning into an educational app bridges foundational computer science concepts with practical use, which is especially relevant for a Computer Science learning project.

5. Support Future Scalability

The enhancement lays the groundwork for additional smart features in the future, such as performance alerts, tailored exercises, or adaptive difficulty.



Updated UI/UX Screenshots (if applicable)

Old User Interface



keinjivelina15@gmail.com

Keinji Velina



Good Evening!

Game On

[Learn More](#)



Stages

Easy

Pronunciation
Reading
Word Pop



Interm

Mystery L
Hidden W
Jumbled

Stories To Read

[View All](#)

The Fox and the Leopard

Aesop's Fables



The Frog and The Ox

Aesop's Fables



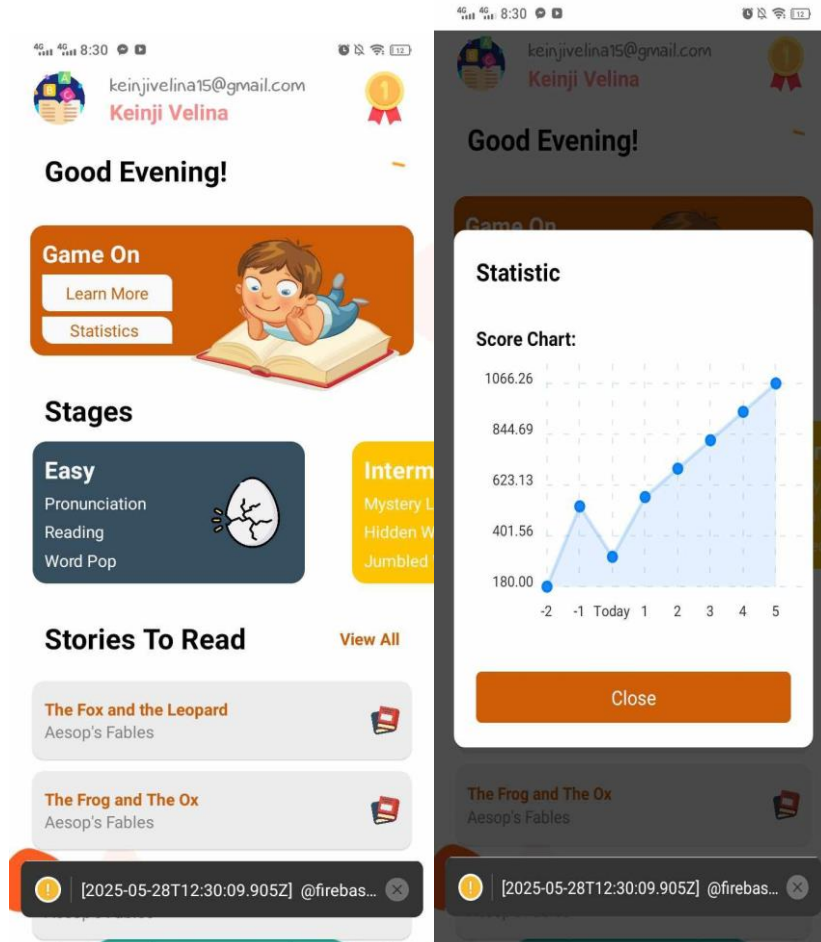
The Fox and The Grapes

Aesop's Fables





Updated User Interface





Testing Approach and Results

Approach	Description	Applied To	Result
Unit Testing	Tested individual functions such as ML model prediction	TensorFlow model, Firebase reads	a. Functions like the TensorFlow prediction return correct values b. Firebase data is retrieved and processed as expected
Integration Testing	Checked if modules communicate correctly	Data flow from Firebase → ML → Chart	c. Data flows properly from Firebase to the ML model d. Predictions update the graph without errors
Functional Testing	Ensured all features behave as expected	Login, Reading Activities, Statistics	e. "Statistic" button triggers prediction f. Login works g. Reading activities update score data h. Graph shows accurate results
Performance Testing	Measured app responsiveness and lag	Chart rendering, model prediction time	i. App responds in average time during predictions and chart updates j. Few noticeable lag or crashes on lower-end devices



Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna



Technologies and Frameworks Used



Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna



Technology	Description
React Native	A JavaScript framework for building cross-platform mobile apps (Android and iOS) using a single codebase.
Expo Go	A development toolchain and client app for building and testing React Native projects without needing native build tools. Speeds up development and testing.
JavaScript / TypeScript	Used as the primary programming language to build the app logic, UI, and ML integration
React Navigation	A routing library used to manage screen transitions within the app (e.g., Home → Statistics screen).
react-native-chart-kit	A library for displaying charts and graphs, used to visualize the student's actual and predicted scores.
Firebase Authentication	Handles user sign-up, sign-in, and secure session management



Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna



Firestore	A real-time NoSQL cloud database used to store and retrieve student data, daily points, and activity results.
TensorFlow.js	A JavaScript library for training and running machine learning models directly in the browser or mobile apps. In GameOn, it's used to build a linear regression model to predict student scores.
TensorFlow.js Core API	Used to define tensors, build the model (sequential()), compile it with a loss function, and train it using .fit().



Developer Notes / Installation Instructions

To use the system do the following:

1. Install Expo Go (SDK 49 only) in your mobile phone.
2. Log in the account in the Expo Go App
 - a) Email: iamtol20
 - b) Password: kenken123
3. Open Visual Studio code and change directory to the GameOn project
4. Open terminal and type these commands in order
 - a) npm install
 - b) npx expo start
5. After the QR code generate, use the Expo Go app to scan the generated QR code in the VS Code.

*after scanning it will load the system which usually takes 1-2 minutes.