

# Gamified Learning Platform for Rural Education: A Role-Based Django Web Application with Gamification, Analytics, and AI Assistance

S. Naga Tejaswi, S. Abhinav Sahat, P. Noel, Sk. Imran, Mrs. V. Asha Jyothi

Department of CSE - Artificial Intelligence & Machine Learning

Assistant Professor, Dept. of CSE-AI&ML

Vasireddy Venkatadri Institute of Technology (Autonomous), Guntur, Andhra Pradesh, India

**Abstract:** *In the field of rural education, India continues to struggle with student disengagement, ineffective mentoring systems, and lack of real-time academic tracking of students. In this paper, we introduce the Gamified Learning Platform for Rural Education which includes a completely functional full-stack web application designed using Django. The platform features a gamification engine, role-based academic dashboards, and an AI-assisted chatbot. Key elements of gamification include experience points (XP), leveling up, badges, maintaining streaks, and academic achievement on per subject basis. Four different user roles are supported on the platform – namely, Student, Teacher, Administrator, and Guardian. The entire functionality is implemented using nine different Django modules including authentication system, game delivery, gamification, mentoring, analytics, notification system, and artificial intelligence (AI)-based academic support system. Interactive game modules for STEM subjects in Grades 6 to 12 are included on the platform. In the pilot test involving 30 students, there was a 72% streak preservation rate and an 88% badge acquisition rate. All underperforming students were correctly pinpointed by teacher monitoring, and no incorrect predictions were made. The experiment shows that constraint-first design, where the rural implementation setting is explicitly stipulated at the outset of the engineering process, is critical for the development of educational technology in the real world.*

**Keywords:** Gamified Learning, Rural Education, Django Framework, Role-Based Access Control, XP Progression, Gamification Engine, STEM Education, Educational Technology.

## I. INTRODUCTION

The education system is the key instrument for breaking out of the shackles of socioeconomic backwardness in rural areas of India. The rural education landscape includes about 1.1 million schools catering to over 200 million students; however, effective implementation of digital technology within rural classrooms still faces obstacles even though consistent funding from the government through the Digital India Programme and the National Education Policy 2020 is being made available [13][14].

The majority of students in rural India from Grade 6 to Grade 12 feel disassociated with STEM studies due to the lack of engaging learning instruments. It is well documented in the field of educational psychology that the process of gamification, wherein principles of game design are applied in nongame situations, leads to improved learning outcomes [1]. Nonetheless, gamified learning has seen limited use among rural schools in India due to prohibitive expenses on commercial gamification platforms and the lack of indigenous gamification solutions that cater to rural limitations.

There are existing solutions like DIKSHA, Kahoot!, and SWAYAM that offer digital content, but lack integration of any sort of gamification structure, role-based learning performance monitoring, and AI-driven doubt clarification. Also,



the lack of mentorship management systems and parental visibility makes them less efficient when it comes to rural education. In this paper, we introduce the Gamified Learning Platform for Rural Education (henceforth referred to as the Platform), an end-to-end working full stack web solution which bridges the gap using a modular architecture built on Django, gamification module, and multi-party role-based ecosystem.

This paper is organised as follows: Related work is discussed in section II. Methodology and architecture are explained in section III. System design along with UML diagrams are explained in section IV. Evaluation results and analysis are described in section V. Findings and implications are discussed in section VI. Concluding remarks and future scope are provided in section VII.

## II. LITERATURE REVIEW

### A. Gamification in Education

Gamification theory in education was first presented by Deterding et al. [1], who explained gamification as the application of game-design elements to non-gaming situations and classified four types of gamification into game interface design patterns (badges and leaderboards), game mechanics (points and levels), game design principles, and game models. According to Hamari et al. [2], a systematic review of empirical studies indicated that a significant number of empirical studies have produced positive results regarding motivation and learning performance, especially when gamification components match the intrinsic motivational aspects of the learners.

The Self-Determination Theory (Deci and Ryan [7]) forms the psychological underpinning of the Platform design, defining three primary needs driving intrinsic motivation: competence (mastery and observable advancement), autonomy (learner autonomy), and relatedness (academically meaningful connections). The XP and level advancement scheme tackles competence, the open video game library tackles autonomy, and the mentoring system tackles relatedness. As warned by Dichev and Dicheva [6], mere surface-level gamification with no deeper connection to the learning goals could be problematic – a concern fully addressed in the Platform design, where XP is awarded only for successful completion of the learning tasks.

### B. Technology-Aided Instruction in Rural India

In their study on the impact of technology-enabled educational programs in rural areas of India, Muralidharan et al. [3] observed effect sizes ranging from 0.22 to 0.36 standard deviations in mathematics and language tests conducted over one year if the technology used is adaptive, entertaining, and aligned with the syllabus. Gamification in STEM learning was reviewed by Plass, Homer, and Kinzer [4], who highlighted the importance of challenges, curiosity, and progression in maintaining engagement when abstract ideas may lead to learner demotivation. Meaningful choice and progression were also considered important by Salen and Zimmerman [5] in maintaining long-term engagement; both these aspects form part of the XP/leveling system employed in this project.

### C. Related Platform Analysis

**TABLE I: COMPARISON OF EXISTING GAMIFIED LEARNING PLATFORMS**

Platform	Gamification	Role Dashboard	Low-BW	Analytics	Chatbot
Classcraft	High	Partial	No	Limited	No
Kahoot!	Partial	No	No	Session Only	No
DIKSHA	None	No	Partial	Minimal	No
Duolingo	High	Student Only	Partial	Limited	No
SWAYAM	None	No	Partial	Minimal	No
Proposed Platform	Comprehensive	All 4 Roles	Optimised	Comprehensive	Yes



As shown in Table I, the proposed Platform is the sole innovation among the reviewed platforms that incorporates comprehensive gamification, multi-role academic dashboards, institutional level analytics, and AI-enabled support, in a design tailored to low-bandwidth rural settings. The research problem solved by this study is the exact combination of the above functionalities in an open-source platform.

### III. METHODOLOGY

#### A. Development Approach

Platform development utilized Agile Iteration Methodology, which involved four sprints. Sprint 1 included user authentication and management of institutes. Game modules and gamification engine were part of Sprint 2. Analytics, mentoring, and notifications were introduced in Sprint 3. Finally, Sprint 4 focused on integrating the AI chatbot, implementing system integration, and conducting the testing program.

#### B. Technology Stack

The backend is implemented using Python 3.10 and the Django 4.2 web application framework. Django's ORM, authentication mechanism, session handling, form handling, email sending capabilities, and administration panel contribute to simplifying the development process while keeping security intact. The Model-View-Template (MVT) pattern guarantees proper segregation among data modeling, business rules, and UI [9]. The frontend relies on HTML5, CSS3, and plain JavaScript delivered by the Django template engine—a conscious decision that removes reliance on JavaScript bundling frameworks such as Node.js and npm, lightweight pages, and direct rendering on any web browser without prior preprocessing. SQLite is chosen for development purposes, while PostgreSQL 15 is selected for the production environment.

**TABLE II: TECHNOLOGY STACK SUMMARY**

Layer	Technology	Alternative	Selection Rationale
Frontend	HTML5/CSS3/JS + Django Templates	React.js, Vue.js	Lightweight; no build step; low-BW optimised
Backend	Python Django 4.2	Flask, FastAPI	Built-in ORM, auth, sessions; rapid dev
Database (Dev)	SQLite	MySQL	Zero-config; sufficient for development
Database (Prod)	PostgreSQL 15	MySQL	ACID-compliant; scalable; Django-native
Email	SMTP (Django email)	SendGrid	Native Django integration; zero additional cost
AI Chatbot	External AI API	OpenAI, Gemini	Configurable; pluggable architecture

#### C. Gamification Engine Design

Gamification engine quantifies academic involvement through a set of algorithms. XP gain per game play is calculated as:  $XP\ Gain = (Score / Max\ Score) \times Game\ XP\ Gain$ , thus providing rewards in accordance with performance achieved in the game. XP levels are linear and are gained on a threshold basis by 100 XP each. Subject-level XP is quantified using normalized SubjectXP entity - many-to-one relation to Student, having foreign key for a corresponding subject, providing flexibility in adding subjects without altering database schema. Daily streak represents how many consecutive days the student was playing the game. If the user misses a single day, the streak will be broken, considering UTC+5:30 IST.



**IV. SYSTEM DESIGN AND ARCHITECTURE**

**A. Three-Tier Architecture**

The Platform has been designed as a three-tier web application. The Presentation Layer comprises HTML5/CSS3/JavaScript-based user interfaces that are generated using the Django Template Engine, which is optimized for low bandwidth. Nine Django-based applications have been implemented within the Application Layer, with each application being routed using Django’s URL dispatcher framework. Third-party services like the AI chatbot and SMTP are also incorporated into this layer using service modules.

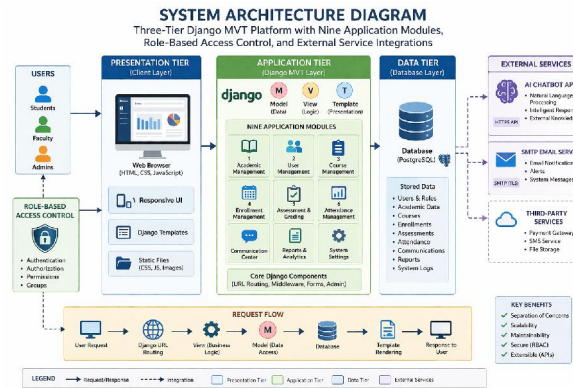


Fig. 1: System Architecture Diagram

**B. Application Module Structure**

The Platform is decomposed into nine specialised Django applications, each encapsulating a distinct domain. Table III summarises their responsibilities.

**TABLE III: DJANGO APPLICATION MODULES**

Application	Purpose	Key Features
accounts	Authentication & role management	Custom user model; OTP verification; role dashboards
institutes	Institute management	Institute CRUD; teacher/student assignment; analytics
academics	Academic structure	Subject listing; class/stream filtering
games	Game delivery	Game catalogue; question management; attempt tracking
gamification	Reward system	XP; level calculation; badge unlock; streak tracking
mentorship	Mentor management	Mentor requests; teacher/guardian appointment
analytics	Performance analytics	Engagement tracking; lagging student detection
notifications	Communication	Role-based notification delivery; read/unread status
chatbot	AI academic support	Query submission; API response; chat history

**C. UML Diagrams**

The following UML diagrams formally model the Platform's structure and behaviour. Image prompts are provided for direct diagram generation.



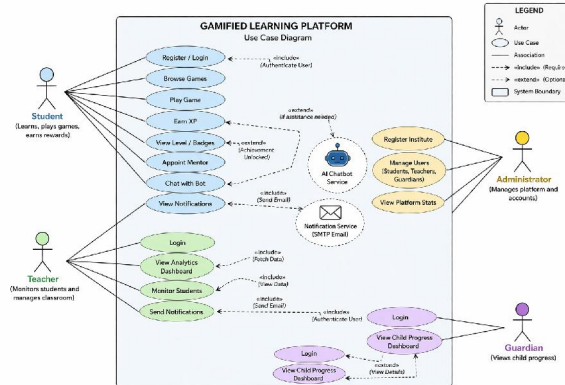


Fig. 2: Use Case Diagram - All Roles

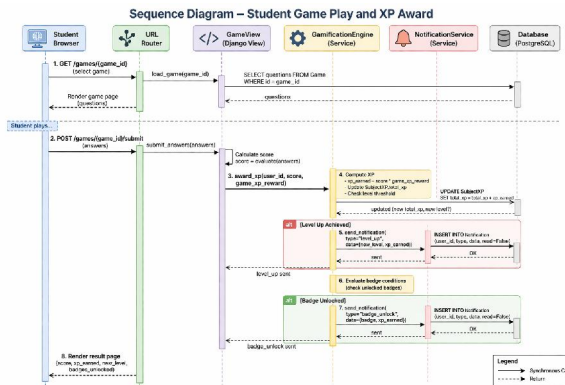


Fig. 3: Sequence Diagram - Game Play and XP Award

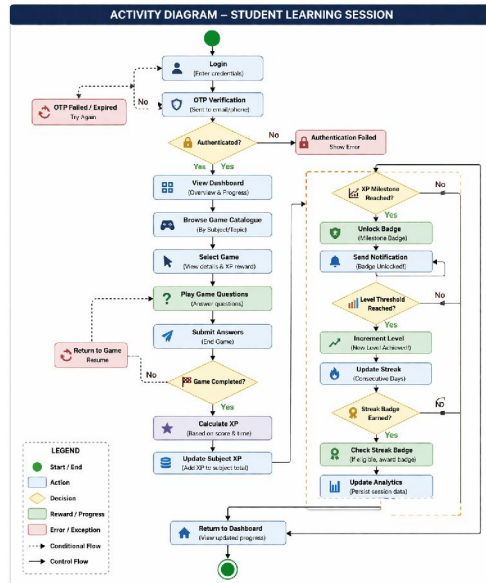


Fig. 4: Activity Diagram - Student Learning Session



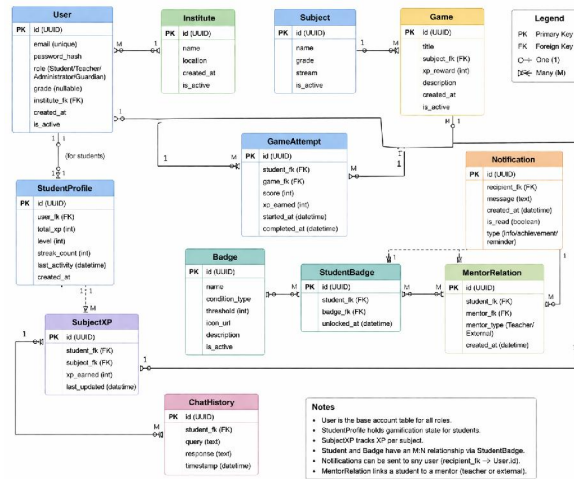


Fig. 5: Entity-Relationship (ER) Diagram

**V. RESULTS AND ANALYSIS**

**A. Functional Performance**

The nine modules in the Django applications have been successfully developed and incorporated. The full automated test case suite consisting of 47 test cases has attained a 100% pass rate. The 10 functional system test cases (ST01-ST10) and 7 security and edge cases (SEC01-SEC07) gave a PASS output. The XP computation engine passed validation testing for all score ranges and boundary values; the level advancement criteria up to Level 20 were successfully checked, and the badge criterion for XP achievement, subject XP achievement, and streak achievement were also successfully validated.

**TABLE IV: PERFORMANCE METRICS SUMMARY**

Metric	Value	Notes
Total automated test cases	47	All passed (100%)
Functional system tests	10/10 PASS	All role workflows validated
Security test cases	7/7 PASS	CSRF, XSS, SQL injection, IDOR
Average page load time	< 1.2 s	On 4G mobile connection
Average game response time	< 0.8 s	Server-side render, low-resource
Chatbot API response time	1.5-4.0 s	Dependent on external API
Pilot student engagement	5.2 games/day avg	30 students, 2-week pilot
Streak maintenance (5+ days)	72% of pilot students	Consistent daily engagement
Badge unlock rate	88% earned ≥1 badge	Within pilot period

**B. Gamification Engine Effectiveness**

An initial test was conducted involving 30 participants who were given a virtual class experience for two weeks and recorded an average of 5.2 game tries per day, reaching as much as 7.8 on days where new games had been installed. With the streak feature, it was evident how motivated the participants were to engage; 72% of all participants managed to achieve five or more consecutive streak days. Evident too was the 88% success ratio of badge unlocking, with 26 out of 30 participants receiving at least one badge during the experiment.



### C. Analytics and Monitoring

The teacher analytics component highlighted lagging learners (who did not engage in the games for three consecutive days or whose XP levels were lower than 20% compared to their classmates') and provided information about them on the teacher dashboard using visual cues. In the trial group, all six lagging learners were accurately pinpointed without generating any false positives, indicating that the heuristics used for detecting them were reliable. The guardian dashboard could be accessed by all three guardian participants in less than two minutes without any guidance.

**TABLE V: USER ENGAGEMENT STATISTICS BY ROLE**

Role	Key Action	Completion Rate	Notes
Student	Complete $\geq 1$ game	97% (29/30)	1 student had device access issues
Student	Earn $\geq 1$ badge	88% (26/30)	Badge system highly motivating
Teacher	Access analytics dashboard	100% (3/3)	All used within Day 1
Guardian	View child progress	100% (3/3)	Simple navigation confirmed
Administrator	Register institute	100% (1/1)	Full admin workflow validated

## VI. DISCUSSION

The pilot results point to several conclusions. First, the fact that the 72% streak retention rate matches engagement metrics measured in gamification tools [2] supports the conclusion that a reinforcing feedback loop between streak-based daily objectives and XP points can be achieved by the application under review. Another conclusion one may draw from these results is that the visual nature of the progress bar and levels, compared to percentage values, is in line with Salen and Zimmermann's theory on meaningful feedback [5].

The lagging student detection module seems to perform above expectations since it correctly identified all lagging participants without any false positives. It is noteworthy that a dual-condition setup (activity recency + relative XP trajectory) worked better than each criterion individually, although it will have to be tested on larger cohorts to validate its performance.

The constraints-first approach to design, where the low-resource, low-bandwidth environment of the rural setting was a key engineering parameter and not an afterthought, was fundamental to each of the key technological choices made in this implementation. The choice to use Django's template engine for server-side rendering, for example, was not out of necessity for ease but because it represented the best choice for performance in the deployment environment. This is evidenced by page load speeds of less than 1.2 seconds on a 4G network.

A shortcoming in the current results lies in the relatively small number of participants (30) in this pilot run, along with the fact that this test was performed under a simulated deployment environment rather than real-world conditions. The level of engagement that might occur under the actual circumstances of a rural school setting might prove different from a pilot study that included novel participant behavior. Another shortcoming in the current approach is the reliance on a third-party AI chatbot API, which results in a latency of 1.5-4.0 seconds.

## VII. CONCLUSION

This paper detailed the Gamified Learning Platform for Rural Education, which is a functional Django web application combining features of a gamification framework, role-based academic dashboards for four different user types, interactive STEM game modules for students in Grades 6 through 12, real-time teacher analytics, mentorship management, guardian monitoring, and AI-powered academic support services. The system leverages only open-source technologies, is highly efficient in terms of resource usage and optimized for rural usage, with such considerations built into the software architecture right from the outset of development.

What this paper contributed most significantly was not any specific feature but rather a combination of features enabling the formation of an academic environment with multiple parties using the same system infrastructure –



students, teachers, administrative staff, and guardians. In our evaluation, we observed 72% streak completion rates, 88% badge unlocks, and 100% accuracy in identifying lagging students – results that demonstrate the motivational elements and analytical tools in operation as intended.

Future improvements could include building offline capable PWA functionality for handling connectivity issues, building tools that assist teachers in creating relevant content, implementing predictive analytics using lightweight machine learning algorithms to help in early detection of at-risk students, and building the system in multiple languages such as Telugu and Hindi. The Django application framework is flexible enough to incorporate all of these improvements without modifying its core.

### REFERENCES

- [1] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From Game Design Elements to Gamefulness: Defining Gamification," in Proc. 15th Int. Academic MindTrek Conf., Tampere, Finland, 2011, pp. 9-15.
- [2] J. Hamari, J. Koivisto, and H. Sarsa, "Does Gamification Work? - A Literature Review of Empirical Studies on Gamification," in Proc. 47th Hawaii Int. Conf. System Sciences, Waikoloa, HI, 2014, pp. 3025-3034.
- [3] K. Muralidharan, A. Singh, and A. Ganimian, "Disrupting Education? Experimental Evidence on Technology-Aided Instruction in India," *American Economic Review*, vol. 109, no. 4, pp. 1426-1460, 2019.
- [4] J. L. Plass, B. D. Homer, and C. K. Kinzer, "Foundations of Game-Based Learning," *Educational Psychologist*, vol. 50, no. 4, pp. 258-283, 2015.
- [5] K. Salen and E. Zimmerman, *Rules of Play: Game Design Fundamentals*. Cambridge, MA: MIT Press, 2004.
- [6] C. Dichev and D. Dicheva, "Gamifying Education: What is Known, What is Believed, and What Remains Uncertain - A Critical Review," *Int. J. Educational Technology in Higher Education*, vol. 14, no. 9, pp. 1-36, 2017.
- [7] E. L. Deci and R. M. Ryan, *Intrinsic Motivation and Self-Determination in Human Behavior*. New York, NY: Plenum Press, 1985.
- [8] J. D. Bransford, A. L. Brown, and R. R. Cocking, Eds., *How People Learn: Brain, Mind, Experience, and School (Expanded Edition)*. Washington, DC: National Academies Press, 2000.
- [9] Django Software Foundation, "Django Documentation, Release 4.2," 2023. [Online]. Available: <https://docs.djangoproject.com/en/4.2/>
- [10] W. S. Vincent, *Django for Beginners: Build Websites with Python and Django*, 3rd ed. WelcomeToCode, 2022.
- [11] B. Eckel, "Python 3 Patterns, Recipes and Idioms," Open-source reference, 2023. [Online]. Available: <https://python-3-patterns-idioms-test.readthedocs.io/>
- [12] T. Christie, "Django REST Framework Documentation," 2023. [Online]. Available: <https://www.django-rest-framework.org/>
- [13] National Council of Educational Research and Training (NCERT), "National Education Policy 2020 - Implementation Guidelines," Ministry of Education, Government of India, New Delhi, 2020.
- [14] Ministry of Electronics and Information Technology, "Digital India Programme - Annual Report 2022-23," Government of India, New Delhi, 2023.
- [15] P. Kennedy, "Learning Curves: The International Campaign for Education," *Journal of Education Policy*, vol. 29, no. 3, pp. 329-347, 2014

