Developing Game with Education

NISHA S
JANAVI CV

JENU SHAKAR FATHIMA AR

BACHELOR OF ENGINEERING

COMPUTER SCIENCE AND ENGINEERING
ADHIYAMAAN COLLEGE OF ENGINEERING (AN AUTONOMOUS INSTITUTION)

DR. M.G.R NAGAR, HOSUR-635130

ABSTRACT

StoryWise is an innovative educational storytelling and quiz-based game designed to make learning engaging, visual, and interactive for children aged 8 to 15. Built using modern web and mobile technologies such as Angular, Node.js, Express, and MongoDB, the platform combines animated storytelling with gamified quizzes to enhance concept understanding and retention. The system provides role-based access for three key user groups: students who explore animated lessons and take quizzes, teachers who create story-based content and track learner performance, and administrators who manage subjects, progress analytics, and leaderboard data. Key features include an animation-driven learning interface, auto-generated quiz modules, real-time performance analytics, rewards and badges system, voice-over narration, and multilingual support for inclusive learning. StoryWise follows a modular architecture with RESTful API integration, responsive UI design, and a NoSQL schema optimized for scalability and quick retrieval of multimedia content. By combining the power of storytelling and gamification, StoryWise aims to make education emotionally engaging, conceptually clear, and digitally accessible — promoting curiosity-driven learning and improving academic outcomes in young learners through immersive visual experiences.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

StoryWise: Educational Quiz Game with Animated Stories is an innovative web-based learning platform designed to make education more engaging, visual, and interactive for children. The main idea behind the project is to teach lessons through animated storytelling and quiz-based learning, helping students understand concepts easily while keeping them entertained. The project focuses on children aged 8 to 1 years, aiming to develop their curiosity, creativity, and problem-solving skills through the power of digital storytelling.

StoryWise combines education, animation, and gamification into a single platform. Each lesson is presented as a short animated story that visually explains a topic. After completing the story, students take a quiz related to the lesson to test their understanding. Correct answers earn points, badges, or rewards turning learning into a fun and motivating experience. This method helps children learn not only by reading or listening but also by watching, interacting, and participating.

The platform is built using modern web technologies. The frontend is developed with Angular, providing a responsive and user-friendly interface. The backend uses Node.js and Express.js to handle server-side operations like authentication, data processing, and quiz management. The MongoDB database stores user information, stories, quiz data, and performance results. This MEAN stack architecture ensures that the system is fast, scalable, and efficient.

Key features of the platform include animated storytelling modules, autogenerated quizzes, voice-over narration, progress tracking,multilingual.

1.2 OBJECTIVE

The main objective of StoryWise: Educational Quiz Game with Animated Stories is to revolutionize the traditional learning process by integrating education with storytelling, animation, and gamification. The project aims to create an interactive and engaging digital learning platform that enhances conceptual understanding, creativity, and curiosity among children aged 8 to 15. By using animated storytelling as a teaching tool, StoryWise transforms complex educational concepts into simple, visual, and relatable narratives that help students

learn through imagination and experience rather than memorization.

One of the core objectives is to make learning enjoyable and interactive. Traditional classroom methods often fail to capture students' attention or maintain consistent engagement. StoryWise addresses this issue by introducing animated lessons combined with quizzes that encourage participation and immediate feedback. Each story concludes with a short assessment that tests comprehension and critical thinking, enabling students to apply what they learned through fun and play-based interaction.

Another major objective is to empower teachers and educators with a digital tool that simplifies lesson delivery and performance tracking. Teachers can create, upload, and manage animated stories related to their subject topics and generate quiz questions that align with learning outcomes. The system's analytics dashboard provides real-time insights into student performance, helping teachers identify areas where students struggle and take timely remedial actions. This data-driven approach enhances the teaching process and promotes continuous learning improvement.

The platform also aims to support personalized and inclusive learning. By providing multilingual content, voice-over narration, and visually appealing interfaces, StoryWise ensures that students from diverse linguistic and cultural backgrounds can learn comfortably. The system encourages self-paced learning, allowing students to revisit stories and quizzes at their convenience, thereby reinforcing understanding.

Technologically, the project seeks to leverage the strengths of the MEAN stack (MongoDB, Express, Angular, Node.js) to build a scalable, responsive, and secure system. Another objective is to design a role-based access structure, ensuring that students, teachers, and administrators each have customized interfaces suited to their functions. Additionally, the inclusion of reward mechanisms, leaderboards, and badges aims to motivate students and create a sense of achievement, promoting healthy competition and sustained interest in learning.

In summary, StoryWise strives to achieve a blend of education, creativity, and technology. Its

ultimate goal is to make learning an enjoyable adventure, transforming knowledge acquisition into a memorable, story-driven experience that inspires children to explore, think critically, and learn with enthusiasm.

CHAPTER 2 LITERATURE SURVEY

[1] Mayer, R. E. (2001). Mayer's Multimedia Learning Theory emphasizes that students learn better from words and pictures than from words alone. When narration is synchronized with visuals, it helps reduce cognitive overload and promotes meaningful learning. StoryWise applies this by combining animated visuals and voiceovers to simplify complex ideas. The segmenting and signaling principles further help learners retain and transfer knowledge effectively.

[2] Bruner, J. (1991). Bruner emphasized the importance of narrative thinking in education, suggesting that stories help learners construct meaning and relate experiences to learning. Through storytelling, abstract concepts become personal and memorable. In StoryWise, educational lessons are framed as stories, enabling children to connect emotionally and understand lessons more deeply. Narratives foster curiosity and empathy among young learners.

[3] Deterding, S. et al. (2011). Deterding and colleagues introduced gamification as the use of game design elements in non-game contexts to enhance engagement. They found that rewards, points, and feedback systems motivate learners effectively. StoryWise implements these principles with points, levels, and quizzes that encourage continuous participation. The gamified approach promotes fun-based learning rather than passive memorization.

[4] Black, P., & Wiliam, D. (1998). Their research on formative assessment highlights that immediate feedback significantly improves learning outcomes. Providing explanations for right and wrong answers supports conceptual understanding. StoryWise incorporates short quizzes with instant feedback to help students reflect and learn from mistakes. This approach also provides teachers with real-time data on Paivio, A. (1986). Paivio's Dual Coding Theory proposes that people process information through both visual and verbal channels. When both are used together, comprehension improves dramatically. StoryWise's animated videos use narration and visuals simultaneously, enhancing dual-channel learning. This integration helps young learners remember information for a longer time and reduces confusion.

[5] Shute, V. J. (2008). Shute's work on adaptive feedback systems suggests that personalized and timely feedback enhances learning efficiency. StoryWise adapts quiz difficulty based on learners' performance levels. The system offers simpler questions for struggling learners and challenging ones for advanced users. This ensures every child progresses at their own pace while maintaining motivation.

[6] Clark, R. C., & Mayer, R. E. (2016). Clark and Mayer stressed the importance of e-learning design principles like coherence, signaling, and redundancy. They found that removing unnecessary visuals and focusing on essential information improves learning effectiveness. StoryWise adopts these guidelines by presenting clear, focused animations and concise storytelling. The interface minimizes distractions and emphasizes learning goals.

[7] Sweller, J. (1988). Sweller's Cognitive Load Theory explains how working memory limitations can hinder learning when content is overly complex. Breaking information into smaller, manageable segments improves understanding. StoryWise divides lessons into short animated stories followed by quizzes to manage cognitive load effectively. This microlearning approach aligns perfectly with young learners' attention spans.

[8] Gee, J. P. (2003). Gee explored how video games promote problem-solving and critical thinking in educational contexts. He argued that interactive environments encourage exploration, hypothesis testing, and persistence. StoryWise integrates similar principles through interactive quizzes and story-driven challenges. Learners engage in decision-making activities that enhance their reasoning skills and curiosity.

[9] Vygotsky, L. S. (1978). Vygotsky's Social Constructivist Theory emphasizes that learning is a social process involving interaction and collaboration. StoryWise can support this by allowing learners to share stories and compete in group-based quizzes. Collaborative storytelling encourages communication and teamwork. Peer learning also boosts confidence and supports collective problem-solving.

[10] Anderson, L. W., & Krathwohl, D. R. (2001). In their Revised Bloom's Taxonomy, Anderson and Krathwohl described cognitive levels from remembering to creating. StoryWise

quizzes and storytelling activities are designed to engage multiple levels—recall, application, and creation. The platform motivates children not only to remember facts but also to apply and narrate them creatively through stories.

[11] Laurillard, D. (2012). Laurillard's Conversational Framework describes how learning involves iterative interaction between teachers, learners, and content. StoryWise reflects this approach by providing interactive learning cycles—story → quiz → feedback → revision. This continuous feedback loop strengthens concept understanding and supports reflective learning practices among students.

[12] Pappas, C. (2015). Pappas discussed mobile learning trends and their benefits for accessibility and flexibility. Mobile-first learning platforms enable children to study anywhere and anytime. StoryWise, being web and mobile-friendly, ensures easy access even on low-end devices. This aligns with the current shift toward flexible and selfpaced digital education.

[13] Bower, M. (2017). Bower's research on technology-mediated learning environments highlights the value of multimedia interactivity in sustaining learner engagement. Interactive platforms increase attention span and comprehension. StoryWise integrates interactive storytelling and clickable elements to keep learners actively involved. The design supports both guided and exploratory learning styles.

[14] Siemens, G. (2005). Siemens proposed Connectivism, a learning theory for the digital age emphasizing learning through networks and digital interactions. StoryWise aligns with this by connecting learners to diverse animated content.

[15] Al-Azawei, A., Serenelli, F., & Lundqvist, K. (2016). Their study on inclusive education and universal design for learning (UDL) underlines the importance of accessibility features. StoryWise includes subtitles, narration, and simplified navigation for children with different abilities. The design ensures that no learner is excluded due to disabilities or language barriers. Inclusive learning improves overall participation and comprehension.

[16] Romero, M., & Usart, M. (2012).Their work on game-based learning demonstrated that educational games increase motivation and foster creative thinking. StoryWise applies this

principle by transforming traditional lessons into story-based adventures. Learners enjoy entertainment while achieving academic goals. This blend of fun and education enhances long-term retention.

[17] Downes, S. (2012). Downes emphasized the importance of open educational resources (OER) and collaborative content creation. StoryWise can integrate OER animations and teacher-contributed quizzes to expand its content library. This promotes continuous improvement and sharing of quality materials. Open platforms make education more sustainable and cost-effective.

[18] Khalil, M. K., & Elkhider, I. A. (2016). Their research on learning analytics discusses how data-driven systems improve instructional design. StoryWise collects quiz performance and engagement data to identify learning patterns. Teachers can use these insights to personalize instruction and address weaknesses. Analytics also guide curriculum updates for better outcomes.

[19] Johnson, L., & Adams Becker, S. (2018). Their report on emerging educational technologies highlights AI and AR as transformative tools in learning. StoryWise can integrate AI-generated quizzes and adaptive storytelling for a personalized experience. In the future, augmented reality could make story interactions more immersive. Such technologies promise to redefine storytelling-based education.

3.1 EXISTING SYSTEM

CHAPTER 3

SYSTEM ANALYSIS

Education is one of the most powerful tools for shaping young minds, yet traditional teaching methods often fail to capture the interest of modern learners. In many classrooms, children are still taught primarily through textbooks, lectures, and written exercises. Although this approach builds discipline, it lacks interactivity and fails to make learning enjoyable or engaging. As a result, students, especially younger ones, tend to lose focus easily and struggle to connect theory with real-life understanding.

Existing digital education platforms have made progress in bridging the gap between teachers and students, but most remain content-based rather than experience-based. Many e-learning websites and apps primarily deliver static materials such as PDFs, recorded lectures, or reading exercises. While these resources provide access to information, they rarely stimulate a child's imagination or curiosity. There is often little to no interactivity, storytelling, or visual engagement that can make learning fun and memorable for children aged 8–15.

Another limitation of current systems is the lack of gamification and animation. Most platforms fail to use animated visuals or story-based learning approaches that could make complex topics easier to understand. Without interactive quizzes, challenges, or visual storytelling, students often lose interest in continuous learning. This leads to low participation rates, poor retention, and minimal long-term understanding of lessons.

Personalization is another major challenge in existing educational systems. Every child has a different pace, interest, and way of understanding concepts, but most platforms provide uniform content to all learners. This "one-size-fits-all" approach neglects individual strengths and weaknesses. Furthermore, limited progress tracking and feedback systems make it difficult for teachers or parents to assess a child's actual performance .

Finally, most current learning systems focus on theoretical delivery rather than creative comprehension. Children learn best when they see, hear, and interact with content — but many educational tools fail to incorporate this multisensory approach. Without animated visuals, voice narration, and interactive quizzes, digital learning remains passive rather than active.

In summary, the existing educational systems — both traditional and digital — often lack interactivity, personalization, storytelling, and motivation. These limitations highlight the need for an advanced platform like StoryWise, which combines animation, gamification, and storytelling to transform learning into an enjoyable, visual, and interactive experience for children.

3.2 PROPOSED SYSTEM

The traditional education system and many existing e-learning platforms often fail to sustain the interest of young learners because they rely heavily on text-based lessons and lack interactive, story-driven learning experiences. To overcome these challenges, a new innovative platform named StoryWise has been proposed. StoryWise integrates animated storytelling, gamification, and interactive quizzes into a unified web-based learning system designed to make education more engaging, visual, and enjoyable for children.

The proposed system transforms conventional teaching into an interactive learning adventure, where lessons are presented as animated stories that simplify complex topics and capture students' imagination. Each story is followed by a quiz designed to reinforce understanding and evaluate learning outcomes. By combining storytelling and assessment, the platform ensures that students not only learn but also retain knowledge effectively. Gamified features such as points, badges, rewards, and leaderboards motivate children to participate actively and continuously.

StoryWise also focuses on personalized learning, allowing each student to learn at their own pace. Based on quiz performance, the system can recommend easier or more challenging lessons, ensuring that no learner is left behind. Teachers can upload educational stories, design quizzes, and track students' progress through an analytical dashboard. Parents can also monitor their child's learning journey, creating a collaborative ecosystem between students, teachers, and parents.

The system architecture of StoryWise includes role-based authentication to ensure security and proper access control. Students, teachers, and administrators each have separate login interfaces with specific privileges. Students can view stories, play quizzes, and check their scores; teachers can manage content and assess student performance; and administrators oversee user management, reports, and platform maintenance.

3.3 PROPOSED SOLUTION

The proposed solution, StoryWise, is designed to overcome the major challenges in today's educational systems — including low student engagement, lack of creativity, minimal interactivity, and poor retention of knowledge. StoryWise introduces an animationbased

storytelling and quiz-driven learning model that transforms traditional education into an exciting and visually rich experience. By turning lessons into animated stories followed by quizzes and rewards, the platform encourages children to participate actively rather than passively consume information. This combination of storytelling and gamification keeps learners motivated and helps them understand concepts more effectively.

StoryWise focuses on making learning both entertaining and educational. Each subject or lesson is represented as a short animated video that explains the topic in a story format, making abstract ideas easier to grasp. After viewing the story, students take an interactive quiz that tests comprehension and critical thinking. The system rewards learners with points, badges, and leaderboard rankings, motivating them to improve continuously. This gamelike learning process not only builds interest but also enhances memory retention and conceptual clarity.

To achieve high interactivity and seamless access, StoryWise is developed as a full-stack web application using modern technologies. The frontend is built using Angular, ensuring a responsive, intuitive, and visually appealing interface suitable for children. The backend is developed with Node.js and Express.js, managing APIs, authentication, and data communication between the client and server. The MongoDB database efficiently stores user profiles, story modules, quiz data, scores, and progress records. This MEAN (MongoDB, Express, Angular, Node.js) stack architecture ensures scalability, speed, and flexibility.

Students can access animated stories, participate in quizzes, earn points, and view their progress.

Teachers can upload story-based content, create quizzes, track student performance, and provide feedback.

3.4 IDEATION & BRAINSTORMING

The ideation and brainstorming phase of **StoryWise** focused on creating an engaging learning platform that combines storytelling with gamified quizzes to improve student participation and understanding. The team analyzed traditional learning challenges and explored ways to make education more interactive, visual, and enjoyable for children aged 8–15.Through group discussions and idea mapping, features like **animated stories**, **auto- generated quizzes**, **performance tracking**, and **reward-based progression** were conceptualized. Brainstorming sessions also emphasized inclusivity, leading to the addition of **voice-over**

narration and **multilingual support** for diverse learners.

The outcome of this phase was a clear, innovative concept — a storytelling-based educational platform that enhances comprehension and curiosity through interactive, gamelike learning experiences.

3.5 PROBLEM SOLUTION FIT

During the analysis of existing educational systems, several key challenges were identified including **low student engagement**, **lack of personalized learning**, **limited teacher parent interaction**, and **inadequate progress tracking mechanisms**. Traditional e-learning platforms often rely on static teaching methods and one-way content delivery, which fail to sustain students' attention and curiosity. Furthermore, most online learning tools provide limited opportunities for collaboration, feedback, and adaptive learning. As a result, students become disengaged, while teachers struggle to monitor performance and provide timely interventions. These issues emphasized the need for an **interactive and motivation-driven educational platform** that blends storytelling, gamification, and analytics.

To address these challenges, **StoryWise** integrates a **reward-based learning system** featuring **points, badges, levels, and leaderboards** that instill a sense of achievement and healthy competition among learners. This gamified approach transforms education from a passive activity into an **immersive and enjoyable experience**, enhancing both motivation and knowledge retention. The inclusion of **real-time data synchronization** ensures that teachers, students, and administrators can instantly access performance reports, progress analytics, and activity logs, supporting continuous learning and timely feedback.

The platform's use of real-time data synchronization and analytics enstransparency and continuous feedback for all stakeholders. Teachers can create story-based modules, monitor class performance, and generate detailed progress reports using the teacher dashboard. Students can track their learning milestones and identify areas of improvement through personalized feedback, while administrators can oversee content, performance trends, and leaderboard data. This interconnected ecosystem supports collaborative and adaptive learning, ensuring that information flow between students, teachers, and administrators remains seamless and up-to-date.

3.6 ARCHITECTURE DESIGN

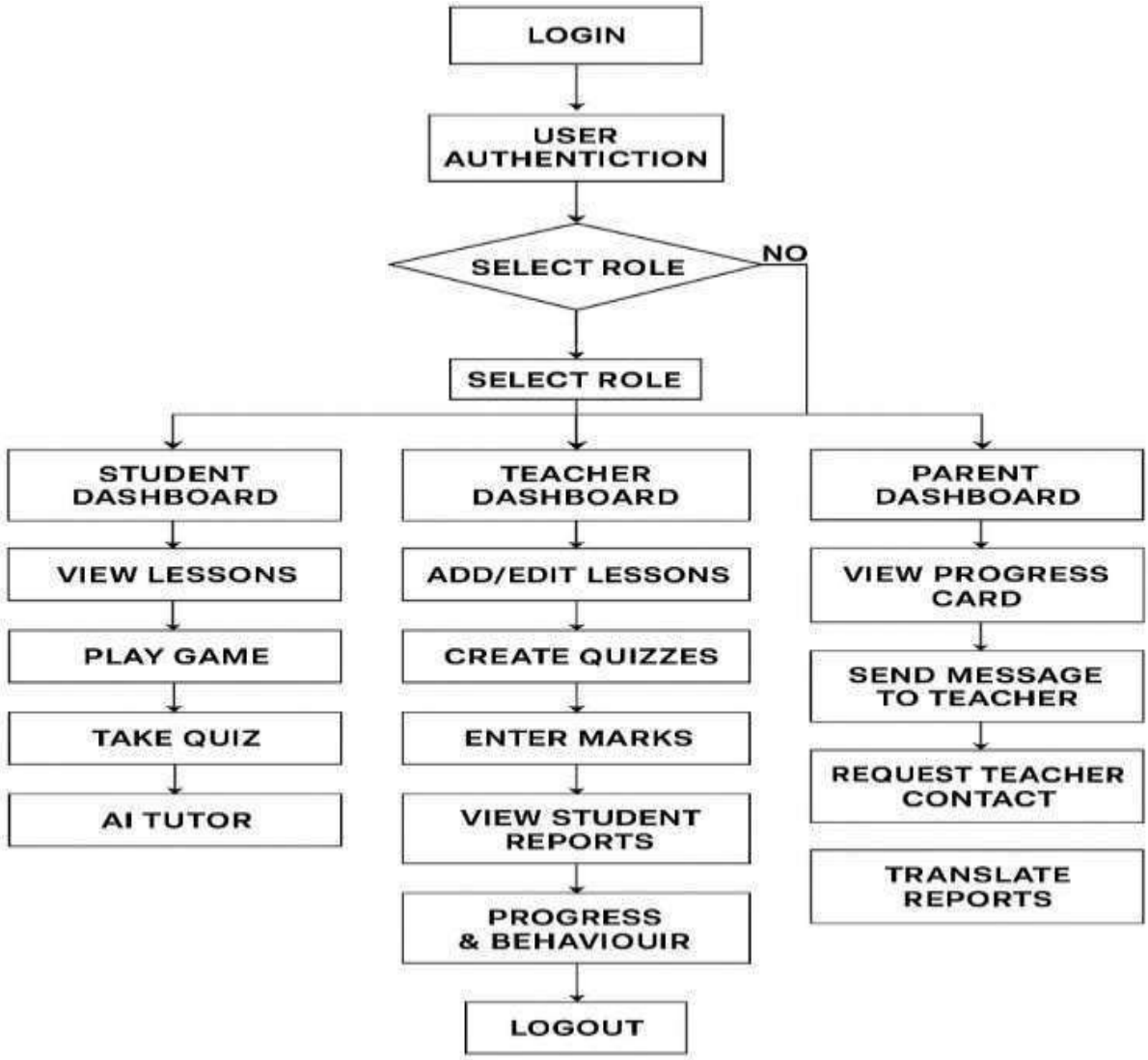

Figure 3.1: Model Architecture

The figure shown above represents the Solution Architecture that we made use of in our Project. In our fitness tracking project, we aim to provide a seamless user experience for tracking exercise routines and achieving fitness goals. Here's a simplified flow of how the project works.

User Input and Data Storage

Users can securely log in to the system and enter relevant information such as exercise names, durations, and completion dates through an intuitive user interface. Each input is validated to ensure data accuracy.

User Authentication and Security

The application implements a robust **user authentication mechanism** safeguard use identities and sensitive data.

Exercise Database and Education

The **Exercise Database serves** as the central repository that stores all exercise-related information, including activity names, durations, categories, and performance metrics.

3.7 DATA FLOW DIAGRAMS

Data Flow Diagrams (DFDs) play a crucial role in understanding and visualizing the movement of data within the **StoryWise Educational Platform**. They provide a clear picture of how information travels between different entities, processes, and data stores within the system. By mapping out data inputs, processing, and outputs, DFDs help in analyzing the functionality and structure.

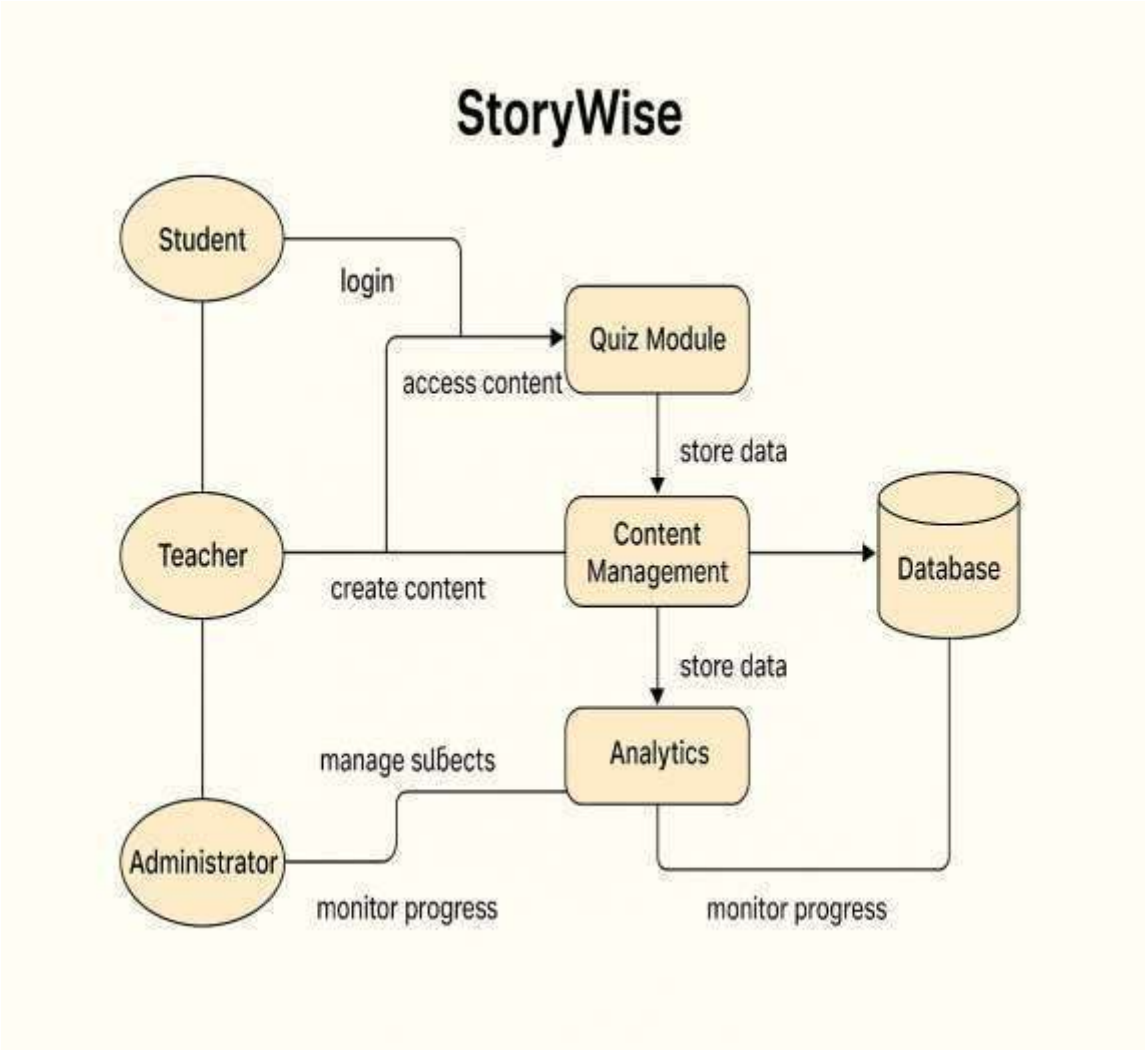

Figure 3.2: Data Flow Diagram

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENT

- **Processor:** Dual-core or higher, 2.0 GHz or above
- **RAM:** Minimum 4 GB (8 GB recommended for smooth operation and multimedia)
- **Hard Disk:** Minimum 20 GB of free space
- **Monitor:** 15" or higher resolution display HD resolution for better visualization of animated lessons
- **Input Devices:** Standard Keyboard and Mouse for user interaction and system control
- **Network:** Stable Internet Connection for online access real-time quiz updates.

4.2 SOFTWARE REQUIREMENTS

- **Operating System:** Windows 10 or later (64-bit) / Linux / macOS
- **Frontend Technologies:** HTML, Tailwind CSS, REACT JSX
- **Backend Language:** React , Express js • **Database:** MySQL
- **Web Server:** Apache Server
- **Browser:** Google Chrome / Microsoft Edge
- **Code Editor:** Visual Studio Code or Sublime Text

CHAPTER 5 IMPLEMENTATION

5.1 DATA COLLECTION

User Authentication

This module handles secure login and registration for students, teachers, and administrators. It uses JWT-based authentication to ensure data security and proper role-based access. Each user role has separate privileges — students can access stories and quizzes, teachers can upload content, and administrators manage the entire system.

Animated Story

This is the core learning module where educational topics are presented as animated storytelling videos. It simplifies complex subjects into engaging visual lessons. Each animation is designed to promote understanding through narration, visuals, and characters that children can relate to.

Quiz and Assessment

After each animated story, students can take an interactive quiz to test their understanding. The module supports multiple-choice questions, score calculation, instant feedback, and correct answer explanations. It enhances learning by reinforcing concepts through active participation..

Gamification and Reward

To make learning fun and motivational, this module includes points, badges, rewards.

Student Dashboard

The student dashboard displays the learner’s progress, quiz scores, badges earned, and learning

history. It provides a personalized view of achievements, helping students track their performance and stay motivated to improve.

Teacher Dashboard

Teachers can upload story content, create quizzes, view student reports, and provide feedback. The dashboard also allows teachers to monitor overall class performance and identify students who need additional support. It bridges the gap between educators and learners.

Admin Management

This module allows administrators to manage user accounts, approve story content, monitor system activities, and generate reports. It ensures smooth operation, content quality, and platform security.

Progress Tracking and Analytics

This module uses collected quiz data and story interactions to generate visual performance reports. Teachers and parents can view charts, graphs, and analytics that show learning trends, strengths, and areas for improvement.

Content Management

This module manages all educational stories, animations, and quizzes uploaded to the platform. It supports adding, editing, and deleting content easily, ensuring that the learning material remains updated, relevant, and age-appropriate.

5.2 COMPONENTS DESIGN

The StoryWise system is designed with modular components to ensure smooth operation and scalability. Each component handles specific tasks such as content delivery, user management, and data processing. The main modules include the Student Module for learning and quizzes, the Teacher Module for content creation and tracking, and the Admin Panel for system control. This structured design enables efficient data flow, easy maintenance, and a seamless interactive experience for all users.

Understand the Project Requirements

The first step focuses on identifying the main goals and functions of the StoryWise platform. It aims to make learning interactive through storytelling, quizzes, and analytics. Key requirements

include secure login, content creation, performance tracking, and real-time notifications. Understanding these needs helps in designing the system's core modules and ensuring smooth interaction between users and components.

Break Down the User Interface

The StoryWise interface is divided into separate modules for students, teachers, and administrators, each with its own dashboard and features. Students can access stories, take quizzes, and view progress; teachers can create content and track performance; administrators can manage users and monitor analytics. This modular interface ensures a clean, user-friendly design and simplifies updates and maintenance.

Plan Component Hierarchy

The StoryWise system follows a well-defined component hierarchy for better organization and scalability. The main parent component manages navigation and routing between student, teacher, and admin dashboards. Each dashboard includes child components for tasks like lesson delivery, quiz management, and performance tracking.

Handle Routing and Navigation

In StoryWise, routing ensures smooth and secure navigation across pages like login, lessons, quizzes, and dashboards. Dynamic routing directs users to their respective roles — students, teachers, or administrators — ensuring access only to authorized sections. This structured navigation provides a seamless and user-friendly experience throughout the platfrom.

Data Flow

In StoryWise, data flows securely between users and the system through well- defined modules. When students interact with lessons or quizzes, the data is sent to the application server for validation and processing before being stored in the MongoDB database. Teachers and administrators access this data for monitoring progress and analytics.

UI/UX Design

The StoryWise interface focuses on simplicity, engagement, and accessibility. Built using Angular and responsive design principles, it ensures smooth performance across all devices. Interactive visuals, animations, and progress indicators make learning enjoyable, while real-time feedback and status updates enhance the overall user experience and keep learners actively engaged.

Accessibility

StoryWise prioritizes accessibility to ensure an inclusive learning experience for all users. The platform uses clear layouts, readable fonts, and well-labeled elements for easy navigation.

Security and Authentication

StoryWise implements role-based authentication to ensure secure access for students, teachers, and administrators. Each user role has specific privileges— students can learn and take quizzes, teachers manage content, and administrators oversee the system**.**

Documentation

All components and modules in StoryWise are thoroughly documented, outlining teir functions, parameters, and interactions. This documentation supports future development, debugging, and system upgrades. It also includes screenshots, workflow diagrams, and usage guidelines to help developers and users understand the system’s structure and functionality clearly.

Scalability

The component design supports scalability by maintaining modularity. New features such as real-time notifications or attendance integration can be added easily. Each module operates independently but communicates effectively through the database and server.

Code Style and Conventions

StoryWise follows consistent and well-structured coding standards to ensure clarity and maintainability. Proper indentation, meaningful naming conventions, and detailed comments are used across all files.

5.3 SOFTWARE DESCRIPTION

HTML

HTML (HyperText Markup Language) is used to structure the web pages of the StoryWise platform. It defines the overall layout and organizes elements such as text, images, animations, buttons, and forms that support storytelling and quizzes. HTML ensures that all content is well-structured, visually clear, and easily accessible to users across devices.

Tailwind CSS

Tailwind CSS is used in StoryWise to design a responsive and clean interface. It styles pages like dashboards, stories, and quizzes, ensuring a modern and userfriendly look across all devices.

JAVA

Java is used in StoryWise to develop the backend logic and RESTful APIs. It handles user requests, processes data, and ensures smooth communication between the frontend and the database.

SPRING BOOT

Spring Boot serves as the backend framework for building and managing the REST APIs in StoryWise. It simplifies development, enhances performance, and ensures secure, efficient server-side operations.

MySQL

MySQL is used in StoryWise to store all essential data, including user information, quiz results, and progress records. It ensures data integrity, security, and fast retrieval, enabling smooth interaction between the backend and the user dashboards.

5.5 RESULT

The StoryWise project successfully delivers an engaging educational platform that combines animated storytelling with interactive quizzes. It provides separate modules for students, teachers, and administrators with secure logins and easy-to-use dashboards. Students learn through fun animated lessons and test their knowledge with quizzes that award points and badges. Teachers can manage content, track performance, and monitor student progress in real

time.

The system ensures smooth data flow, quick access, and strong security using modern technologies like React.js, Node.js, and MongoDB. StoryWise makes learning more enjoyable, visual, and interactive, helping students understand concepts better while staying motivated.

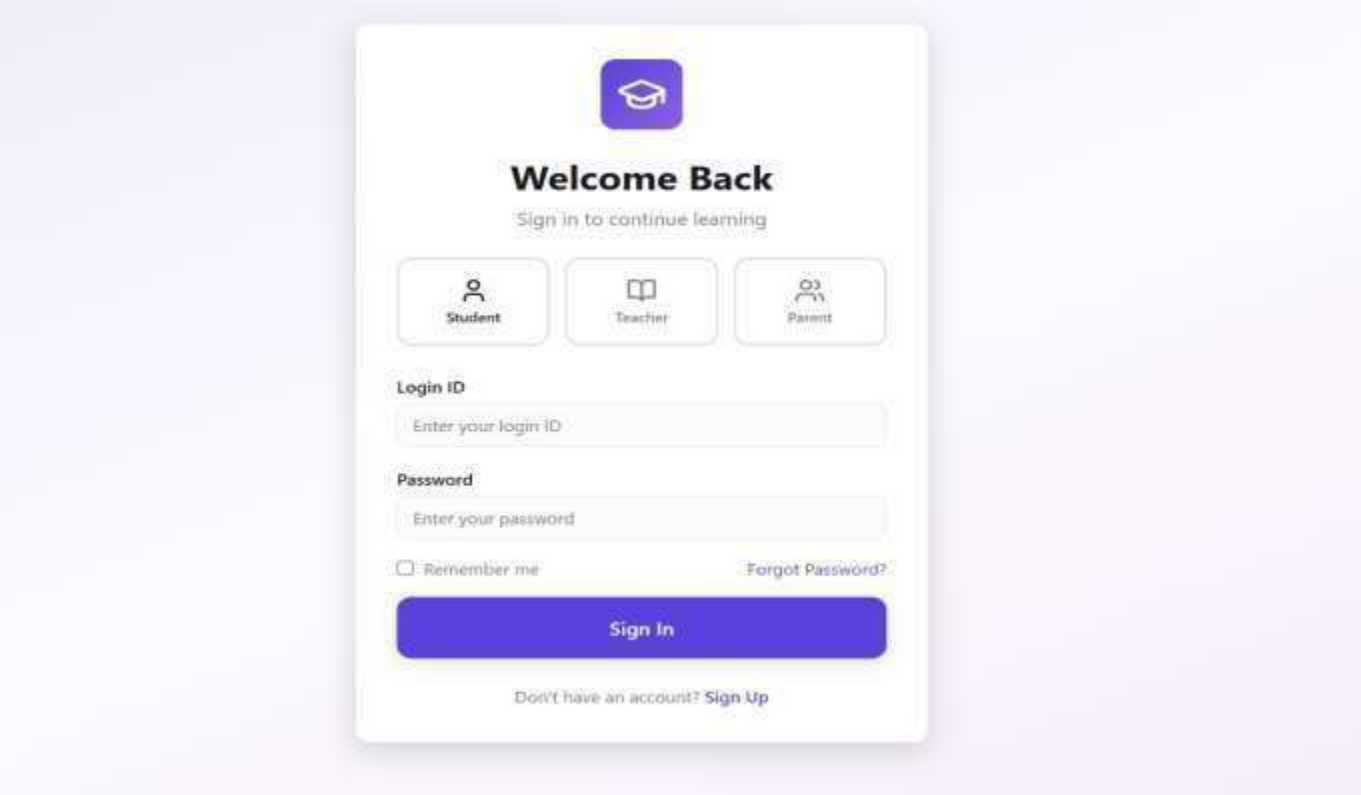

Figure 5.1: Login Page

Figure 5.2: Home page

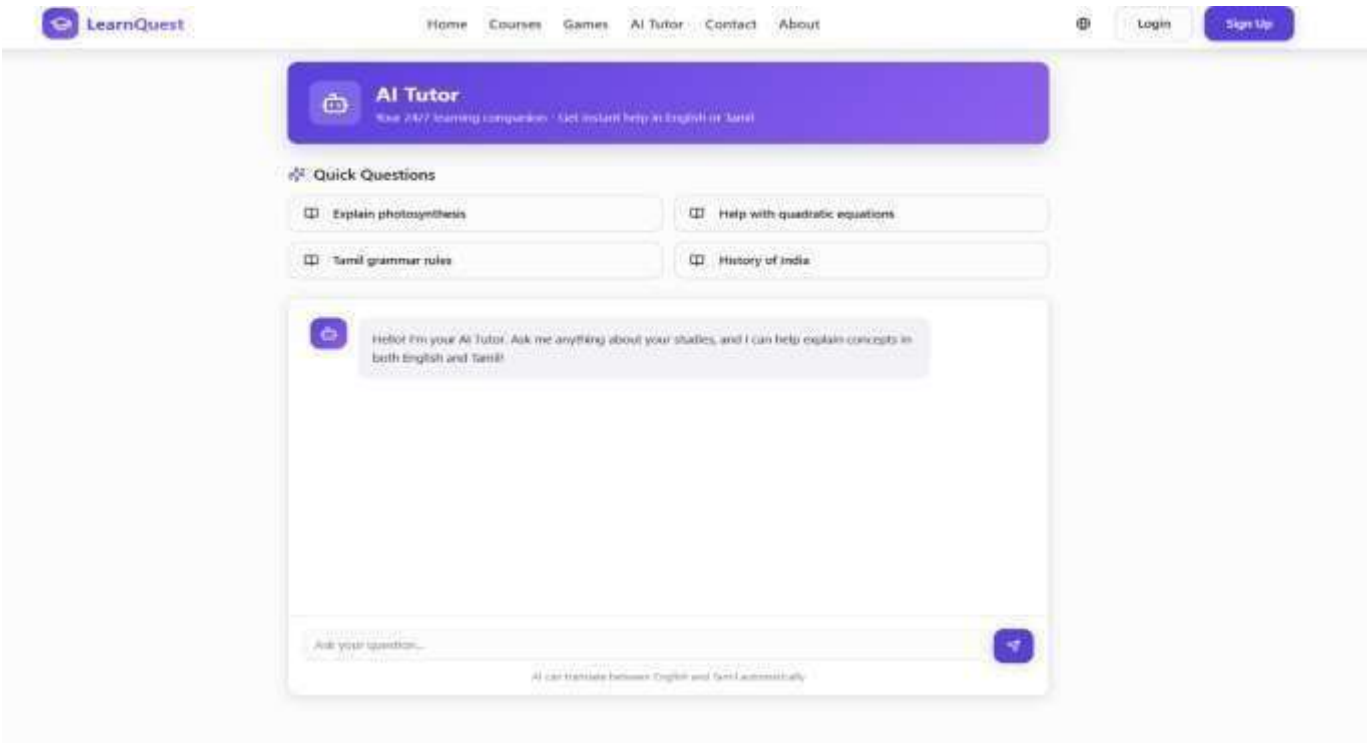

Figure 5.3: AI tutor

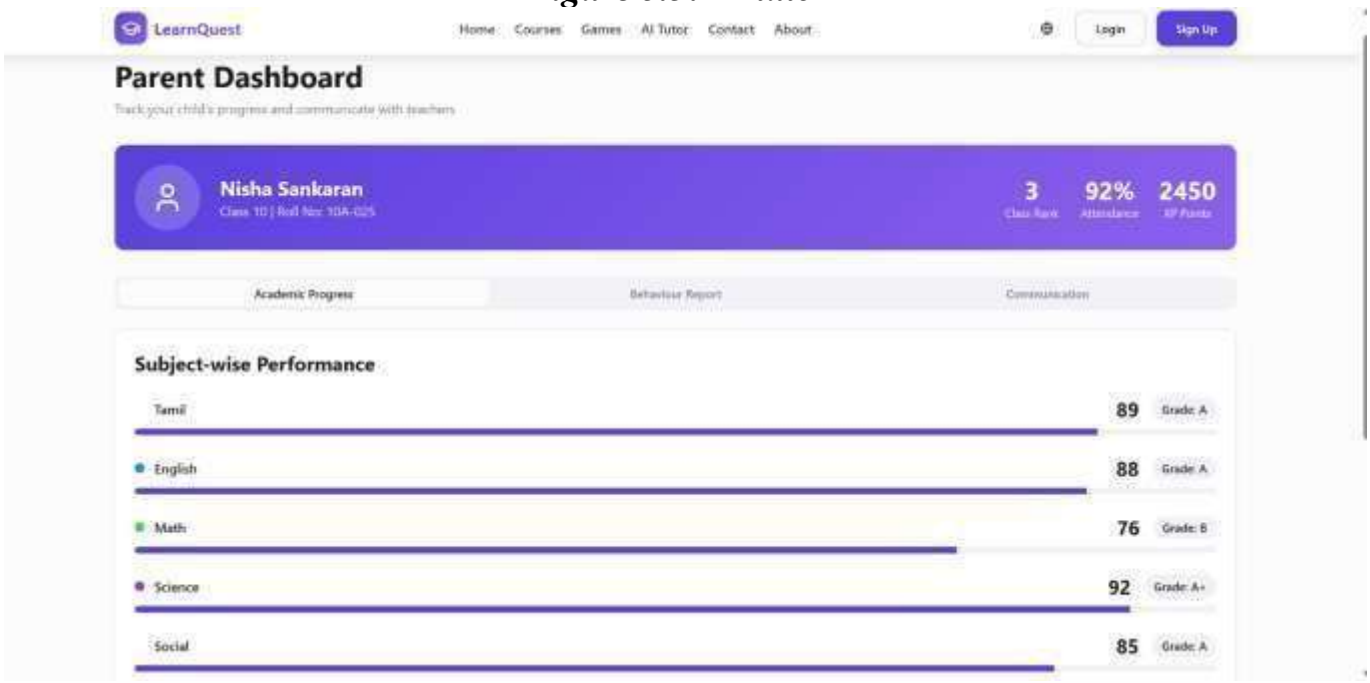

Figure 5.4: Parent Dashboard

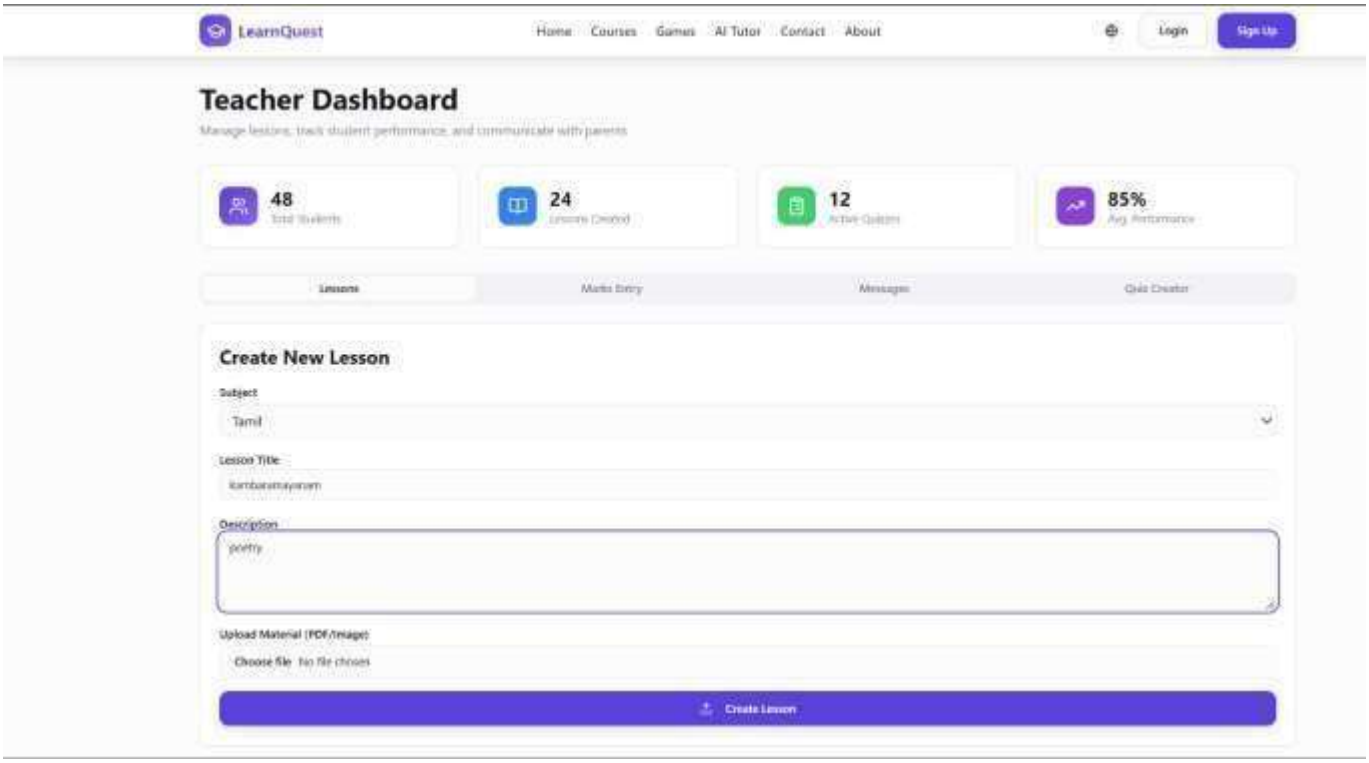

Figure 5.5: Teacher Dashboard

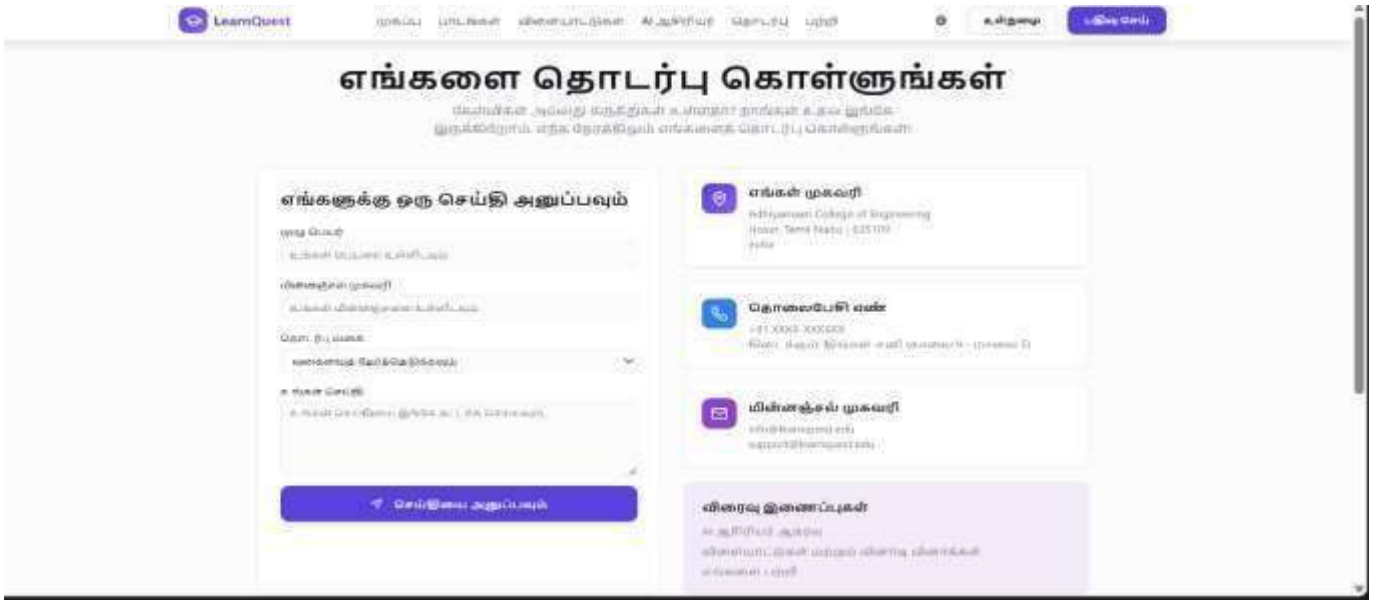

Figure 5.6 : English to Tamil Translate

Figure 5.7: Video

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The StoryWise project successfully combines storytelling and gamification to create an engaging digital learning experience for children. Through animated stories and interactive quizzes, it makes education more visual, fun, and easy to understand. The platform helps students learn actively rather than memorizing lessons passively.

StoryWise includes separate modules for students, teachers, administrators, each with secure logins and simple dashboards. Students can watch stories, take quizzes, and earn rewards, while teachers manage content and track progress. Administrators oversee the overall system and ensure smooth operation.

Developed using React.js, Node.js, and MongoDB, the platform ensures good performance, data security, and scalability. Overall, StoryWise achieves its goal of transforming traditional learning into an enjoyable, interactive, and effective experience that encourages creativity,

motivation, and better understanding among students.

FUTURE SCOPE

The future scope of StoryWise is vast, with numerous opportunities to enhance learning through advanced technologies and expanded features. In the coming versions, the platform can include AI-based personalized learning paths that adapt quizzes and stories according to each student's progress and performance. This will ensure that every learner receives content suited to their level and interests.

The integration of voice interaction and speech recognition can make the platform more engaging, especially for younger learners who may prefer speaking over typing. Adding multilingual support will also help reach a wider audience and promote inclusive learning.

Future enhancements may include teacher analytics dashboards to monitor student performance trends and parent access modules for progress tracking. Deployment on mobile apps will provide students with 24/7 learning access anytime, anywhere.

By incorporating AR/VR storytelling, cloud storage, and gamified rewards, StoryWise can evolve into a fully immersive educational ecosystem that promotes creativity, curiosity, and continuous learning.

APPENDICES

SOURCE CODE

Index.html

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>LearnQuest - Interactive Learning Platform for Classes 8-10</title>
<meta name="description" content="Learn through play with LearnQuest - an interactive educational platform featuring games, AI tutoring, and progress tracking for students in classes 8-10. Available in English and Tamil." />
<meta name="author" content="LearnQuest" />
```

<meta property="og:title" content="LearnQuest - Interactive Learning Platform" />

<meta property="og:description" content="Transform education through game-based learning, AI tutoring, and interactive lessons for classes 8-10" />

<meta property="og:type" content="website" />

<meta property="og:image" content="https://lovable.dev/opengraph- imagep98pqg.png" />

<meta name="twitter:card" content="summary_large_image" />

<meta name="twitter:site" content="@lovable_dev" />

<meta name="twitter:image" content="https://lovable.dev/opengraph- imagep98pqg.png" />

</head>

<body>

<script type="module" src="/src/main.tsx"></script>

</body>

</html>

App.css

.app {

max-width: 1280px; margin: 0 auto; padding: 2rem;

text-align: center;

}

.logo { height: 6em;

padding: 1.5em; will-change: filter;

transition: filter 300ms;

}

.logo:hover {

filter: drop-shadow(0 0 2em #646cffaa);

}

.logo.react:hover {

filter: drop-shadow(0 0 2em #61dafbaa);

}

@keyframes logo-spin { from {

transform: rotate(0deg);

```
}
to {
transform: rotate(360deg);
}
}
@media (prefers-reduced-motion: no-preference) { a:nth-of-type(2) .logo {
animation: logo-spin infinite 20s linear;
}
}
.card { padding: 2em;
}
.read-the-docs { color: #888;
}Index.css @tailwind base;
@tailwind components; @tailwind utilities;
/* Definition of the design system.
All colors, gradients, fonts, etc. should be defined here. All colors MUST be in HSL format.
*/
@layer base {
:root {
/* Gradients */
--gradient-hero: linear-gradient(135deg, hsl(250 70% 55%) 0%, hsl(260 80% 65%) 100%);
--gradient-card: linear-gradient(180deg, hsl(0 0% 100%) 0%, hsl(240 33% 99%) 100%);
/* Animations */
/* Shadows */
}
.dark {
}
}
@layer base {
* {
```

```
@apply border-border;
}
body {
@apply bg-background text-foreground antialiased;
}
h1, h2, h3, h4, h5, h6 { @apply font-bold;
}
}
@layer utilities {
.gradient-hero {
background: var(--gradient-hero);
}
.gradient-card {
background: var(--gradient-card);
}
.shadow-soft {
box-shadow: var(--shadow-soft);
}
.shadow-medium {
box-shadow: var(--shadow-medium);
}
.shadow-strong {
box-shadow: var(--shadow-strong);
}
.shadow-glow {
box-shadow: var(--shadow-glow);
}
.transition-smooth {
transition: var(--transition-smooth);
}
```

```
.transition-bounce {
transition: var(--transition-bounce);
}
/* New utilities */
.gradient-accent {
background: linear-gradient(135deg, hsl(var(--accent)) 0%, hsl(var(--primary-glow)) 100%);
}
.gradient-galaxy {
background: radial-gradient(1200px 600px at 10% -10%, hsl(var(--primary-glow) / 0.25), transparent 60%),
radial-gradient(1200px 600px at 90% 110%, hsl(var(--accent) / 0.25), transparent 60%),
hsl(var(--background));
}
.shadow-neon {
box-shadow: 0 0 0 1px hsl(var(--primary) / 0.2),
0 8px 24px hsl(var(--primary) / 0.25);
}
.animate-bounce-slow { animation: bounce 2.5s infinite;
}
}
```

About.tst

```
import { useState } from "react";
import { Navbar } from "@/components/Navbar"; import { Card } from "@/components/ui/card";
import { Target, Users, Lightbulb, Award } from "lucide-react"; import { Link } from "react-router-dom";
const About = () => {
const [language, setLanguage] = useState("en"); const features = [
{
icon: Target,
title: language === "en" ? "Our Mission" : "எங்கள் ந ோக்கம்", description:
language === "en"
```

? "To transform education through gamification and AI, making learning fun and accessible for every student."
: " கமிபிநகஷன் மற்றும் AI மூலம் கல்வியை மோற்றி, கற்றயல சுவோரஸ்ைமோகவும் எளிதோகவும் ஆக்குவF.",
},
{
icon: Lightbulb,
title: language === "en" ? "Our Vision" : "எங்கள் போர்யவ", description:
language === "en"
? "Creating a world where students enjoy learning and parents stay connected with progress."
: "மோணவர்கள் கற்றயல விரும்பும் மற்றும் பபற்நறோர்கள் முன்நனற்றத்Fடன் இயண ்திருக்கும் உலகம் உருவோக்குதல்.",
},
{
icon: Users,
title: language === "en" ? "Who We Serve" : " ோங்கள் ைோருக்கு நேயவ பேை்கிநறோம்",
description:
language === "en"
? "We serve Students, Teachers, and Parents for a connected education ecosystem."
: "மோணவர்கள், ஆசிரிைர்கள் மற்றும் பபற்நறோர்களுக்கோக இயண ்த கல்வி சூழயல உருவோக்குகிநறோம்.",
},
{
icon: Award,
title: language === "en" ? "Our Values" : "எங்கள் மதிப்புகள்", description:
language === "en"
? "Excellence, innovation, inclusivity, and continuous improvement."
: "சிறப்பு, புFயம, இயணப்பு மற்றும் பதோடர்்சிைோன
return (

<Navbar language={language} setLanguage={setLanguage} />

<h1 className="text-4xl font-bold mb-4">
{language === "en" ? "About LearnQuest" : "LearnQuest பற்றி"}
</h1>
<p className="text-muted-foreground mb-12 max-w-3xl">
{language === "en"

? "LearnQuest is an AI-powered learning platform for grades 8–10 combining games, quizzes, and progress tracking."
: "LearnQuest என்பF 8–10 ஆம் வகுப்பு மோணவர்களுக்கோன AI அடிப்பயடயிலோன கற்றல் தளம் ஆகும்; இதில் வியளோட்டுகள், வினோடி வினோக்கள் மற்றும் முன்நனற்ற கண்கோணிப்பு ஒருங்கியணக்கப்பட்டுள்ளன."}

{features.map((f, i) => (
<Card key={i} className="p-6 hover:shadow-medium transition-smooth">

<f.icon className="h-6 w-6 text-white" />

<h3 className="text-lg font-bold mb-2">{f.title}</h3>
{f.description}
</Card>
))}

<Card className="p-8 bg-gradient-to-br from-primary/10 to-accent/10 mb-12">
<h2 className="text-2xl font-bold mb-4 text-center">
{language === "en"
? "Built with Modern Technology"
: " வீன பதோழில்நுட்பத்Fடன் உருவோக்கப்பட்டF"}
</h2>

{["React", "Tailwind CSS", "AI Integration", "Secure Auth"].map((tech, i) => (

{tech}

{language === "en"
? ["Frontend", "Styling", "Tutoring", "Security"][i]
: ["முன்புறம்", "வடிவயமப்பு", "ஆசிரிைர் வழிகோட்டல்", "போFகோப்பு"][i]}

))}

</Card>

<h2 className="text-2xl font-bold mb-3">
{language === "en" ? "Ready to Transform Learning?" : "கற்றயல மோற்ற தைோர்?"}

```
</h2>
<p className="text-muted-foreground mb-5">
{language === "en"
? "Join students, teachers, and parents shaping the future of education."
: "கல்வியின் எதிர்கோலத்யத வடிவயமக்கும் மோணவர்கள், ஆசிரிைர்கள் மற்றும் பபற்நறோர்களுடன் நேருங்கள்."}
</p>
<div className="flex gap-4 justify-center">
<Link to="/signup">
<button className="px-6 py-3 gradient-hero text-white rounded-lg font-semibold hover:opacity-90">
{language === "en" ? "Get Started" : "பதோடங்குங்கள்"}
</button>
</Link>
<Link to="/contact">
<button className="px-6 py-3 border-2 border-primary text-primary rounded-lg font-semibold hover:bg-primary hover:text-white">
{language === "en" ? "Contact Us" : "எங்கயள பதோடர்புபகோள்ளுங்கள்"}
</button>
</Link>
</div>
</div>
</div>
</div>
);
};
export default About;
```

contact.tsx

```
import { useState } from "react";
import { Navbar } from "@/components/Navbar"; import { Card } from "@/components/ui/card"; import { Button } from "@/components/ui/button"; import { Input } from "@/components/ui/input"; import { Textarea } from "@/components/ui/textarea"; import { Label } from "@/components/ui/label";
import { Mail, Phone, MapPin, Send } from "lucide-react"; import { toast } from "sonner";
import { Link } from "react-router-dom"; const Contact = () => {
const [language, setLanguage] = useState("en");
const [formData, setFormData] = useState({ name: "", email: "", message: "" }); const handleSubmit = (e: React.FormEvent) => {
```

```
e.preventDefault();
if (!formData.name || !formData.email || !formData.message) { toast.error(language === "en" ?
"Please fill all fields" : "அயனத்F புலங்கயளயும் ிரப்பவும்");
return;
}

toast.success(language === "en" ? "Message sent successfully!" : "பேை்தி
பவற்றிகரமோக அFப்பப்பட்டF!");
setFormData({ name: "", email: "", message: "" });
};

return (
<div className="min-h-screen bg-background text-left">
<Navbar language={language} setLanguage={setLanguage} />
<div className="container mx-auto px-4 pt-24 pb-16">
<div className="mb-12">
<h1 className="text-4xl font-bold mb-4">
{language === "en" ? "Contact Us" : "எங்கயள பதோடர்பு பகோள்ளுங்கள்"}
</h1>

? "Have questions or feedback? Reach out to us anytime!"
: "நகள்விகள் அல்லF கருத்Fகள் உள்ளதோ? எங்கயள எப்நபோFம் பதோடர்பு
பகோள்ளலோம்!"}
</p>
</div>
<div className="grid md:grid-cols-2 gap-8 max-w-5xl mx-auto">
{/* Contact Form */}
<Card className="p-8 text-left">
<h2 className="text-2xl font-bold mb-6">
{language === "en" ? "Send us a Message" : "எங்களுக்கு ஒரு பேை்தி
அFப்பவும்"}
</h2>
<form onSubmit={handleSubmit} className="space-y-4">
<div>
<Label>{language === "en" ? "Full Name" : "முழு பபயர்"}</Label>
<Input type="text"
placeholder={language === "en" ? "Enter your name" : "உங்கள் பபயர
உள்ளிடவும்"}
value={formData.name}
onChange={(e) => setFormData({ ...formData, name: e.target.value })}
/>
```

```
</div>
<div>
<Label>{language === "en" ? "Email Address" : "மின்னஞ்ேல் முகவரி"}</Label>
<Input type="email"
placeholder={language === "en" ? "Enter your email" : "உங்கள் மின்னஞ்ேயல உள்ளிடவும்"}
value={formData.email}
onChange={(e) => setFormData({ ...formData, email: e.target.value })}
/>

</div>
<div>
<Label>{language === "en" ? "Message" : "பேைச்தி"}</Label>
<Textarea
placeholder={language === "en" ? "Type your message..." : "உங்கள் பேைச்தியை எழுதவும்..."}
rows={4} value={formData.message}
onChange={(e) => setFormData({ ...formData, message: e.target.value })}
/>

</div>
<Button type="submit" className="w-full" size="lg">
<Send className="mr-2 h-4 w-4" />
{language === "en" ? "Send Message" : "பேைச்தி அFப்பவும்"}
</Button>
</form>
</Card>
{/* Contact Info */}
<div className="space-y-6 text-left">
<Card className="p-6 flex items-start gap-4">
<div className="gradient-hero p-3 rounded-lg">
<MapPin className="h-6 w-6 text-white" />
</div>
<div>
<h3 className="font-bold mb-2">
{language === "en" ? "Address" : "முகவரி"}
</h3>
<p className="text-muted-foreground"> Adhiyamaan College of Engineering<br /> Hosur, Tamil Nadu, India
</p>
</div>
```

Teacherdashboard.tsx

```
import { useState } from "react"; import { Navbar } from "@/components/Navbar"; import { Card } from "@/components/ui/card";
import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input"; import { Label } from "@/components/ui/label"; import { Textarea } from "@/components/ui/textarea";
import { Tabs, TabsList, TabsTrigger, TabsContent } from "@/components/ui/tabs";
import { BookOpen, Upload, MessageSquare, ClipboardList, Users, TrendingUp } from "lucide-react"; const TeacherDashboard = () => { const [language, setLanguage] = useState("en");
const students = [
{ id: 1, name: "Nisha", grade: "10", tamil: 89, english: 88, math: 76, science: 92 },
{ id: 2, name: "Raj", grade: "9", tamil: 78, english: 82, math: 88, science: 75 },
];
const messages = [
{ id: 1, from: "Parent of Nisha", subject: "Math concern", date: "2025-01-15" },
{ id: 2, from: "Parent of Raj", subject: "Request meeting", date: "2025-01-14" },
];
return (

<div className="min-h-screen bg- background text-left">
<Navbar language={language} setLanguage={setLanguage} />
<div className="container mx-auto px-4 pt-24 pb-16">
<h1 className="text-3xl font-bold mb- 2">
{language === "en" ? "Teacher Dashboard" : "ஆசிரிைர் டோஷ்நபோர்டு"}
</h1>
<p className="text-muted-foreground mb-8">
{language === "en"
? "Manage lessons, marks, and communicate with parents"
: "போடங்கள், மதிப்பபண்கள் மற்றும் பபற்நறோர்களுடன் பதோடர்பு
```

பகோள்ளுங்கள்"}

{/* Stats */}

{[["Students", "மோணவர்கள்", 48,

Users], ["Lessons", "போடங்கள்", 24, BookOpen],

["Quizzes", "வினோடி வினோ", 12, ClipboardList], ["Avg", "ேரோேரி", "85%", TrendingUp]]

.map(([t, ta, n, Icon], i) => (

<Card key={i} className="p-5 flex items-center gap-3">

<Icon className="h-6 w-6 text-primary"

/>

{n}

{language === "en" ? t : ta}

</Card>

))}

{/* Tabs */}

<Tabs defaultValue="lessons">

<TabsList className="grid grid-cols-4 w-full">

<TabsTrigger value="lessons">{language

=== "en" ? "Lessons" : "போடங்கள்"}</TabsTrigger>

<TabsTrigger value="marks">{language

=== "en" ? "Marks" : "மதிப்பபண்கள்"}</TabsTrigger>

<TabsTrigger value="messages">{language === "en" ? "Messages" :

"பேை்திகள்"}</TabsTrigger>

<TabsTrigger value="quiz">{language

=== "en" ? "Quiz" : "வினோடி வினோ"}</TabsTrigger>

```
</TabsList>
{/* Lessons */}
<TabsContent value="lessons">
<Card className="p-6">
<h2 className="text-2xl font-bold mb- 3">
{language === "en" ? "Create Lesson" : "போடத்யத உருவோக்கவும்"}
</h2>
<Label>{language === "en" ? "Title" : "தயலப்பு"}</Label>
<Input placeholder={language === "en"
? "Enter title" : "தயலப்யப
உள்ளிடவும்"} className="mb-3" />
<Label>{language === "en" ? "Description" : "விளக்கம்"}</Label>
<Textarea rows={3} placeholder={language === "en" ? "Lesson details" : "போட விளக்கம்"}
className="mb-3" />
<Label>{language === "en" ? "Upload File" : "நகோப்யப பதிநவற்றவும்"}</Label>
<Input type="file" className="mb-3" />
<Button className="w-full"><Upload className="mr-2 h-4 w-4" />{language
=== "en" ? "Save" : "நேமிக்கவும்"}</Button>
</Card>
</TabsContent>
{/* Marks */}
<TabsContent value="marks">
<Card className="p-6 overflow-x- auto">
<h2 className="text-2xl font-bold mb- 3">{language === "en" ? "Student Marks" :
"மோணவர் மதிப்பபண்கள்"}</h2>
<table className="w-full">
<thead><tr><th>Name</th><th>Grade</ th><th>Tamil</th><th>English</th><th
>Math</th><th>Science</th></tr></thea d>
<tbody>
```

```
{students.map((s) => (
<tr key={s.id} className="border-b">
<td className="p- 2">{s.name}</td><td>{s.grade}</td><td
>{s.tamil}</td>
<td>{s.english}</td><td>{s.math}</td>
<td>{s.science}</td>
</tr>
))}
</tbody>
</table>
</Card>
</TabsContent>
{/* Messages */}
<TabsContent value="messages">
<Card className="p-6">
<h2 className="text-2xl font- bold mb-3">{language === "en" ? "Parent
Messages":"பபற்நறோர்
பேை்திகள்"}</h2>
{messages.map((m) => (
<Card key={m.id} className="p-3 flex justify-between items-center mb-3">
<div>
<p className="font- semibold">{m.from}</p>
<p className="text- sm">{m.subject}</p>
<p className="text-xs text-muted- foreground">{m.date}</p>
</div>
<Button size="sm">{language === "en"
? "Reply" : "பதில்"}</Button>
</Card>
))}
</Card>
```

```
</TabsContent>
</Tabs>
</div>
</div>
);
};
export default TeacherDashboard;
```

Gameplay.tsx

```
import { useParams, Link, useNavigate } from "react-router-dom"; import { useState, useEffect } from "react";
import { Card } from "@/components/ui/card"; import { Button } from "@/components/ui/button"; import { Badge } from "@/components/ui/badge";
import { ArrowLeft, Trophy, Sparkles, CheckCircle2 } from "lucide-react"; import { games } from "@/data/games";
import { toast } from "sonner"; export default function Gameplay() { const { id } = useParams();
const navigate = useNavigate();
const game = games.find((g) => g.id === id); const [isPlaying, setIsPlaying] = useState(false);
const [isCompleted, setIsCompleted] = useState(false); const [index, setIndex] = useState(0);
const [score, setScore] = useState(0);
const [selected, setSelected] = useState<number | null>(null);
const questions: Record<string, { q: string; options: string[]; answer: number }[]> = { Tamil: [
{ q: "“அறம் பேை விரும்பு” எFம் பபோருள்?", options: ["பபருயம", " ல்லF பேைஃ", "ேண்யட", "தூங்கு"], answer: 1 },
{ q: "தமிழ் உயிபரழுத்F எF?", options: ["க்", "இ", "ங்", "த்"], answer: 1 },
{ q: "திருக்குறள் எழுதிைவர்?", options: ["அவ்யவைோர்", "கம்பர்", "திருவள்ளுவர்", "போரதி"], answer: 2 },
],
English: [
{ q: "Pick the adjective: “The quick fox jumps.”", options: ["fox", "quick", "jumps", "the"],
```

answer: 1 },

{ q: "Past tense of “go” is…", options: ["goed", "going", "went", "gone"], answer: 2 },

{ q: "Synonym of “happy”?", options: ["sad", "glad", "angry", "late"], answer: 1 },

],

Math: [

{ q: "7 × 8 = ?", options: ["54", "56", "58", "64"], answer: 1 },

{ q: "Solve: 2x = 10", options: ["2", "4", "5", "10"], answer: 2 },

{ q: "Sum of triangle angles?", options: ["90°", "120°", "180°", "270°"], answer: 2 },

],

Science: [

{ q: "H2O is…", options: ["Oxygen", "Hydrogen", "Water", "CO2"], answer: 2 },

{ q: "Force unit?", options: ["Joule", "Newton", "Watt", "Pascal"], answer: 1 },

{ q: "Plants make food by…", options: ["Respiration", "Photosynthesis", "Digestion", "Evaporation"], answer: 1 },

],

Social: [

{ q: "Capital of India?", options: ["Mumbai", "Chennai", "Kolkata", "New Delhi"], answer: 3 },

{ q: "Father of Nation?", options: ["Nehru", "Gandhi", "Patel", "Ambedkar"], answer: 1 },

{ q: "Tropic of Cancer passes through?", options: ["North", "South", "Middle", "No"], answer: 2 },

],

Computer: [

{ q: "CPU stands for?", options: ["Central Processing Unit", "Computer Personal Unit", "Control Power Unit", "Central Program Unit"], answer: 0 },

{ q: "Python is a…", options: ["Hardware", "OS", "Programming language", "Game"], answer: 2 },

{ q: "Shortcut to copy?", options: ["Ctrl+V", "Ctrl+C", "Ctrl+X", "Ctrl+Z"], answer: 1 },

],

};

useEffect(() => { if (!game) {

toast.error("Game not found"); navigate("/games");

}

```
}, [game, navigate]); if (!game) return null;
const startGame = () => { setIsPlaying(true); setIsCompleted(false); setIndex(0);
setScore(0); setSelected(null);
};
const handleSelect = (i: number) => { if (selected !== null) return; setSelected(i);
if (i === questions[game.subject][index].answer) setScore((s) => s + 1);
setTimeout(() => {
if (index < questions[game.subject].length - 1) { setIndex((n) => n + 1);
setSelected(null);
} else { setIsPlaying(false); setIsCompleted(true);
toast.success(`🎉 You earned ${game.xpReward} XP!`, { description: `Score:
${score}/${questions[game.subject].length}`,
});
}
}, 600);
};
return (
<div className="min-h-screen bg-gradient-to-b from-indigo-900 to-purple-900">
<div className="container py-8 text-left">
<Link to="/games">
<Button variant="ghost" className="mb-6">
<ArrowLeft className="mr-2 h-4 w-4" /> Back
</Button>
</Link>
{/* Start Screen */}
{!isPlaying && !isCompleted && (
<Card className="p-8 text-left">
<h1 className="text-3xl font-bold mb-3">{game.name}</h1>
<p className="text-muted-foreground mb-6">{game.description}</p>
<Badge>{game.subject}</Badge>
<Button onClick={startGame} className="w-full mt-6 gradient-hero text-white">
```

```
<Sparkles className="mr-2 h-5 w-5" /> Start Game
</Button>
</Card>
)}
{/* Quiz Screen */}
{isPlaying && (
<Card className="p-8 text-left">
<h2 className="text-xl font-bold mb-4"> Q{index + 1}: {questions[game.subject][index].q}
</h2>
<div className="grid gap-3">
{questions[game.subject][index].options.map((opt, i) => { const correct = i ===
questions[game.subject][index].answer; const selectedOpt = selected === i;
return (
<Button key={i}
variant="outline"
className={`justify-start h-auto py-3 ${ selected !== null
? correct
? "border-green-500 text-green-600"
: selectedOpt
? "border-red-500 text-red-500"
: ""
: ""
}`}
onClick={() => handleSelect(i)} disabled={selected !== null}
>
{opt}
</Button>
);
})}
</div>
```

```
<p className="mt-4 text-sm">Score: {score}</p>
</Card>
)}
{/* Completed Screen */}
{isCompleted && (
<Card className="text-center p-8">
<Trophy className="mx-auto h-20 w-20 text-yellow-400 mb-4" />
<h2 className="text-3xl font-bold mb-2">Congratulations!</h2>
<p className="mb-4">You completed {game.name}</p>
<div className="text-2xl font-bold mb-4 text-green-500">{game.xpReward} XP Earned!</div>
<Button onClick={startGame} className="w-full mb-3 gradient-hero text-white">
<Sparkles className="mr-2 h-5 w-5" /> Play Again
</Button>
<Link to="/games">
<Button variant="outline" className="w-full">
<ArrowLeft className="mr-2 h-5 w-5" /> Back to All Games
</Button>
</Link>
</Card>
)}
</div>
</div>
);
}
```

login.tsx

```
import { useState } from "react";
import { useNavigate } from "react-router-dom"; import { Button } from "@/components/ui/button"; import { Card } from "@/components/ui/card"; import { Input } from "@/components/ui/input"; import { Label } from "@/components/ui/label";
import { GraduationCap, User, BookOpen, Users } from "lucide-react"; import { toast } from
```

```
"sonner";
const Login = () => {
const [userType, setUserType] = useState<"student" | "teacher" | "parent">("student"); const
[loginId, setLoginId] = useState("");
const [password, setPassword] = useState(""); const navigate = useNavigate();
const handleLogin = (e: React.FormEvent) => { e.preventDefault()
if (!loginId || !password) { toast.error("Please fill in all fields"); return;
}
toast.success(Welcome back, ${userType}!); if (userType === "student") {
navigate("/student/dashboard");
} else if (userType === "teacher") { navigate("/teacher/dashboard");
} else { navigate("/parent/dashboard");
}
};
const userTypes = [
{ type: "student" as const, icon: User, label: "Student", color: "subject-english" },
{ type: "teacher" as const, icon: BookOpen, label: "Teacher", color: "subject-science" },
{ type: "parent" as const, icon: Users, label: "Parent", color: "subject-social" }
];
return (
<div className="min-h-screen bg-gradient-to-br from-primary/10 via-background to-
accent/10 flex items-center justify-center p-4">
<Card className="w-full max-w-md shadow-strong">
<div className="p-8">
{/* Logo */}
<div className="flex justify-center mb-6">
<div className="gradient-hero p-3 rounded-xl">
<GraduationCap className="h-10 w-10 text-white" />
</div>
</div>
<h1 className="text-3xl font-bold text-center mb-2">Welcome Back</h1>
```

```
{/* User Type Selection */}
<div className="grid grid-cols-3 gap-3 mb-6">
{userTypes.map(({ type, icon: Icon, label, color }) => (
<button key={type}
onClick={() => setUserType(type)}
className={`p-4 rounded-lg border-2 transition-smooth ${ userType === type
? border-${color} bg-${color}/10
: "border-border hover:border-primary/50"
}`}
<Icon className={`h-6 w-6 mx-auto mb-1 ${userType === type ? text-${color} : "text- muted-foreground"}`} />
<p className={`text-xs font-semibold ${userType === type ? text-${color} : "text- muted-foreground"}`}>
{label}
</p>
</button>
))}
</div>
{/* Login Form */}
<form onSubmit={handleLogin} className="space-y-4">
<div>
<Label htmlFor="loginId">Login ID</Label>
<Input id="loginId" type="text"
placeholder="Enter your login ID" value={loginId}
onChange={(e) => setLoginId(e.target.value)} className="mt-1"
/>
</div>
<div>
<Label htmlFor="password">Password</Label>
<Input id="password" type="password"
placeholder="Enter your password" value={password}
```

```
onChange={(e) => setPassword(e.target.value)} className="mt-1"
/>
</div>
<div className="flex items-center justify-between text-sm">
<label className="flex items-center gap-2 cursor-pointer">
<input type="checkbox" className="rounded" />
<span className="text-muted-foreground">Remember me</span>
</label>
<a href="#" className="text-primary hover:underline"> Forgot Password?
</a>
</div>
<Button type="submit" className="w-full" size="lg"> Sign In
</Button>
</form>
<div className="mt-6 text-center text-sm text-muted-foreground"> Don't have an account?{"
"}
<button
onClick={() => navigate("/signup")}
className="text-primary font-semibold hover:underline"
>
Sign Up
</button>
</div>
</div>
</Card>
</div>
);
};
```

Home.tsx

```
import { useState } from "react";
```

```
import { Navbar } from "@/components/Navbar";
import { CourseCard } from "@/components/CourseCard"; import { Button } from "@/components/ui/button";
import { BookOpen, Gamepad2, Bot, Trophy, Users, Sparkles } from "lucide-react"; import heroBanner from "@/assets/hero-banner.jpg";
import tamilImg from "@/assets/subject-tamil.jpg"; import englishImg from "@/assets/subject-english.jpg"; import mathImg from "@/assets/subject-math.jpg"; import scienceImg from "@/assets/subject-science.jpg"; import socialImg from "@/assets/subject-social.jpg";
import computerImg from "@/assets/subject-computer.jpg"; import { useNavigate, Link } from "react-router-dom"; const Home = () => {
const [language, setLanguage] = useState("en"); const navigate = useNavigate();
const courses = [
{
title: language === "en" ? "Tamil" : "தமிழ்",
description: language === "en" ? "Learn Tamil with Thirukkural Games" : "திருக்குறள் வியளோட்டுகளுடன் தமிழ் கற்றுக்பகோள்ளுங்கள்",
image: tamilImg,
color: "hsl(var(--subject-tamil))", path: "/games"
},
{
title: language === "en" ? "English" : "ஆங்கிலம்",
description: language === "en" ? "Improve English with fun stories" : "நவடிக்யனைகோன கயதகளுடன் ஆங்கிலம் நமம்படுத்Fங்கள்",
image: englishImg,
color: "hsl(var(--subject-english))", path: "/games"
},
{
title: language === "en" ? "Mathematics" : "கணிதம்",
description: language === "en" ? "Solve puzzles to unlock levels" : " ியலகயளத் திறக்க புதிர்கயளத் தீர்க்கவும்",
```

image: mathImg,

color: "hsl(var(--subject-math))", path: "/games"

},

{

title: language === "en" ? "Science" : "அறிவிைல்",

description: language === "en" ? "Perform experiments virtually" : "பமை ிகர் நேோதயனகயளே் பேை்யுங்கள்",

image: scienceImg,

color: "hsl(var(--subject-science))", path: "/games"

},

{

title: language === "en" ? "Social Studies" : "ேமூக அறிவிைல்",

description: language === "en" ? "Explore history & geography" : "வரலோறு மற்றும் புவியிையல ஆரோயுங்கள்",

image: socialImg,

color: "hsl(var(--subject-social))", path: "/games"

},

{

title: language === "en" ? "Computer Science" : "கணினி அறிவிைல்",

description: language === "en" ? "Learn coding the fun way" : "நவடிக்யகைோன வழியில் குறியீடுகயள கற்றுக்பகோள்ளுங்கள்",

image: computerImg,

color: "hsl(var(--subject-computer))", path: "/games"

}

];

const features = [

{

icon: Gamepad2,

title: language === "en" ? "Game-Based Learning" : "வியளோட்டு அடிப்பயடயிலோன கற்றல்",

description: language === "en" ? "Learn by playing interactive games" : "ஊடோடும் வியளோட்டுகள் மூலம் கற்றுக்பகோள்ளுங்கள்"

},

{

icon: Bot,

title: language === "en" ? "AI Tutor Support" : "AI ஆசிரிைர் ஆதரவு",

description: language === "en" ? "Get help instantly anytime" : "எப்நபோF நவண்டுமோனோலும் உடனடிைோக உதவி பபறுங்கள்"

},

{

icon: Users,

title: language === "en" ? "Teacher & Parent Connect" : "ஆசிரிைர் மற்றும் பபற்நறோர் இயணப்பு",

description: language === "en" ? "Track progress and communicate easily" : "முன்நனற்றத்யதக் கண்கோணிக்கவும் எளிதோகத் பதோடர்புபகோள்ளவும்"

}

];

return (

<Navbar language={language} setLanguage={setLanguage} />

{/* Hero Section */}

<section

className="relative pt-24 pb-16 px-4 overflow-hidden" style={{

backgroundImage: url(${heroBanner}), backgroundSize: 'cover', backgroundPosition: 'center'

}}

>

<Sparkles className="h-4 w-4" />

{language === "en" ? "The Future of Fun Learning" : "நவடிக்யகைோன கற்றலின் எதிர்கோலம்"}

<h1 className="text-5xl md:text-6xl font-bold mb-6 leading-tight">

{language === "en" ? "Learn. Play. Achieve." : "கற்றல். வியளைோடு. ோோதிக்க."}

</h1>

{language === "en"

? "An interactive platform for classes 8-10 with games, lessons, AI tutor, and progress tracking."

: "வியளைோட்டுகள், போடங்கள், AI ஆசிரிைர் மற்றும் முன்நனற்ற கண்கோணிப்புடன் வகுப்புகள் 8-10க்கோன ஊடோடும் தளம்."}

<Button variant="hero" size="lg"

onClick={() => navigate("/signup")}

>

<BookOpen className="mr-2 h-5 w-5" />

{language === "en" ? "Start Learning" : "கற்றல் பதோடங்கு"}

</Button>

<Button size="lg" variant="outline"

className="bg-white/10 border-white/30 text-white hover:bg-white/20 backdrop-blur- sm"

onClick={() => {

document.getElementById('courses')?.scrollIntoView({ behavior: 'smooth' });

}}

>

```
<Trophy className="mr-2 h-5 w-5" />
{language === "en" ? "Browse Subjects" : "போடங்கயள உலோவவும்"}
</Button>
</div>
</div>
</div>
{/* Animated floating elements */}
<div className="absolute top-20 left-10 animate-bounce">
<div className="w-16 h-16 bg-white/10 backdrop-blur-sm rounded-full" />
</div>
<div className="absolute bottom-20 right-10 animate-bounce delay-150">
<div className="w-12 h-12 bg-white/10 backdrop-blur-sm rounded-full" />
</div>
</section>
{/* Features Section */}
<section className="py-16 px-4 bg-secondary/30">
<div className="container mx-auto">
<h2 className="text-3xl md:text-4xl font-bold text-center mb-12">
{language === "en" ? "Why LearnQuest?" : "ஏன் LearnQuest?"}
</h2>
<div className="grid md:grid-cols-3 gap-8 max-w-5xl mx-auto">
{features.map((feature, index) => (
<div key={index}
className="bg-card p-6 rounded-xl shadow-soft hover:shadow-medium transition-smooth text-center"
>
<div className="gradient-hero w-16 h-16 rounded-full flex items-center justify-center mx-auto mb-4">
<feature.icon className="h-8 w-8 text-white" />
</div>
<h3 className="text-xl font-bold mb-2">{feature.title}</h3>
```

```
<p className="text-muted-foreground">{feature.description}</p>
</div>
))}
</div>
</div>
</section>
{/* Courses Section */}
<section id="courses" className="py-16 px-4">
<div className="container mx-auto">
<div className="text-center mb-12">
<h2 className="text-3xl md:text-4xl font-bold mb-4">
{language === "en" ? "Explore Our Courses" : "எங்கள் போடத்திட்டங்கயள ஆரோயுங்கள்"}
</h2>
<p className="text-muted-foreground text-lg max-w-2xl mx-auto">
{language === "en"
? "Interactive lessons designed for classes 8-10 with gamified content"
:"வியளோட்டு உள்ளடக்கத்Fடன் வகுப்புகள் 8-10க்கோக வடிவயமக்கப்பட்ட ஊடோடும் போடங்கள்"}
</p>
</div>
<div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6 max-w-6xl mx-auto">
{courses.map((course, index) => (
<CourseCard key={index} {...course} />
))}
</div>
</div>
</section>
{/* Footer */}
<footer className="bg-card border-t border-border py-12 px-4">
<div className="container mx-auto">
```

<h3 className="font-bold text-lg mb-4">LearnQuest</h3>

{language === "en"

? "Transforming education through play and AI"

: "வியளோட்டு மற்றும் AI மூலம் கல்வியை மோற்றுதல்"}

<h4 className="font-semibold mb-3">{language === "en" ? "Quick Links" : "வியரவு இயணப்புகள்"}</h4>

<ul className="space-y-2 text-sm text-muted-foreground">

<Link to="/" className="hover:text-primary transition-smooth">{language === "en" ? "Home" : "முகப்பு"}</Link>

<Link to="/#courses" className="hover:text-primary transition-smooth">{language === "en" ? "Courses" : "போடங்கள்"}</Link>

<Link to="/games" className="hover:text-primary transition-smooth">{language === "en" ? "Games" : "வியளோட்டுகள்"}</Link>

<h4 className="font-semibold mb-3">{language === "en" ? "Support" : "ஆதரவு"}</h4>

<ul className="space-y-2 text-sm text-muted-foreground">

<Link to="/contact" className="hover:text-primary transition-smooth">{language === "en" ? "Contact" : "பதோடர்பு"}</Link>

<Link to="/about" className="hover:text-primary transition-smooth">{language === "en" ? "About Us" : "எங்கயள பற்றி"}</Link>

<h4 className="font-semibold mb-3">{language === "en" ? "Connect" : "இயணக்கவும்"}</h4>

{language === "en" ? "Follow us on social media" : "ேமூக ஊடகங்களில் எங்கயளப் பின்பதோடருங்கள்"}

© 2025 LearnQuest. {language === "en" ? "All rights reserved." : "அயனத்F உரியமகளும் போFகோக்கப்பட்டயவ."}

</footer>

);

};

REFERENCES

[1]. Gee, J. P. (2003). What Video Games Have to Teach Us About Learning and Literacy. New York: Palgrave Macmillan.

[2]. Prensky, M. (2001). Digital Game-Based Learning. McGraw-Hill.

[3]. Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design Elements to Gamefulness: Defining “Gamification”. Proceedings of the 15th International Academic MindTrek Conference.

[4]. Squire, K. (2011). Video Games and Learning: Teaching and Participatory Culture in the Digital Age. Teachers College Press.

[5]. Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does Gamification Work? – A Literature Review of Empirical Studies on Gamification. Proceedings of the 47th Hawaii International

Conference on System Sciences.

[6]. Bandura, A. (1986). Social Foundations of Thought and Action: A Social Cognitive Theory. Prentice-Hall. Bandura, A. (1986). Social Foundations of Thought and Action: A Social Cognitive Theory. Prentice-Hall.

[7]. Mayer, R. E. (2009). Multimedia Learning (2nd ed.). Cambridge University Press.

[8]. Papastergiou, M. (2009). Digital Game-Based Learning in High School Computer Science Education: Impact on Educational Effectiveness and Student Motivation. Computers & Education, 52(1), 1–12.

[9]. Kapp, K. M. (2012). The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education. Pfeiffer.

[10]. Zichermann, G., & Cunningham, C. (2011). Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps. O'Reilly Media.