

GuhuzaQuiz - Technical Documentation

Table of Contents

1. [Introduction](#)
2. [Web Application Overview](#)
3. [System Architecture](#)
4. [Technical Stack](#)
5. [Core Features](#)
6. [Database Design](#)
7. [Installation Guide](#)
8. [Deployment Options](#)
9. [Security Considerations](#)

Introduction

GuhuzaQuiz is an advanced interactive learning platform designed to provide users with a progressive learning experience through quizzes. The application combines modern web technologies with gamification elements to create an engaging educational environment.

Web Application Overview

User Interface and Experience

1. Landing Page

- Welcome screen with user authentication options
- Featured quizzes and top performers
- Quick start guide for new users
- Latest achievements and community highlights

2. Dashboard

- Personal progress overview
- Quick access to ongoing quizzes
- Achievement badges display
- Daily challenges and rewards
- Personalized learning recommendations

3. Quiz Interface

- Clean, distraction-free design
- Real-time progress indicator
- Timer display for timed questions
- Instant feedback on answers
- Score calculation and display
- Celebration animations for achievements

Core Functionalities

1. Quiz Management

- **Topic Selection**

- 10 diverse subject areas
- History, Science, Geography, Mathematics
- Literature, Technology, Sports, Arts
- General Knowledge, Current Affairs

- **Difficulty Levels**

- Level 1: Beginner (Basic concepts)
- Level 2: Elementary (Foundation building)
- Level 3: Intermediate (Complex concepts)
- Level 4: Advanced (Critical thinking)
- Level 5: Expert (Mastery level)

- **Question Types**

- Multiple choice questions
- True/False statements
- Fill in the blanks
- Match the following
- Short answer questions

2. Learning Progress

- **Progress Tracking**

- Topic-wise completion rates
- Difficulty level progression
- Time spent per topic
- Success rate analysis
- Performance trends

- **Achievement System**

- Topic mastery badges
- Speed achievement rewards
- Consistency streaks
- Special challenge completions
- Community contributions

3. Social Features

- **Leaderboards**

- Global rankings
- Topic-specific leaderboards
- Weekly/Monthly competitions
- Friend challenges
- Achievement showcases

- **Community Engagement**

- Friend connections
- Group competitions
- Discussion forums
- Question suggestions

- Peer reviews

4. User Profiles

- **Profile Management**
 - Personal information
 - Learning preferences
 - Achievement showcase
 - Statistics dashboard
 - Friend connections
- **Customization Options**
 - Interface themes
 - Notification settings
 - Privacy controls
 - Learning path preferences
 - Quiz difficulty adjustments

5. Administrative Features

- **Content Management**
 - Question bank management
 - Topic organization
 - Difficulty calibration
 - User feedback integration
 - Content quality control
- **User Management**
 - Account administration
 - Role assignments
 - Access control
 - Activity monitoring
 - Support system

System Architecture

Frontend Architecture

- **Framework:** Next.js 14 with React 18
- **Styling:** Tailwind CSS for responsive design
- **State Management:** React Context API
- **Animations:** Framer Motion
- **Type Safety:** TypeScript implementation

Backend Architecture

- **API Routes:** Next.js API routes
- **Database ORM:** Prisma
- **Authentication:** NextAuth.js
- **Database:** MySQL
- **Real-time Features:** Server-side events and WebSocket connections

Technical Stack

Frontend Technologies

- Next.js 14
- React 18
- TypeScript
- Tailwind CSS
- Framer Motion
- Canvas Confetti (for celebrations)
- React Icons

Backend Technologies

- MySQL Database
- Prisma ORM
- NextAuth.js
- Node.js (v18+)

Development Tools

- Docker for containerization
- Git for version control
- npm for package management

Core Features

Quiz System

1. Progressive Learning

- 50 distinct levels
- 5 difficulty tiers
- Adaptive difficulty scaling

2. Question Engine

- Dynamic question generation
- Timer-based challenges
- Multiple question types
- Real-time scoring

3. Performance Tracking

- Individual statistics
- Progress metrics
- Achievement system
- Historical data analysis

User Management

1. Authentication System

- Secure login/registration
- OAuth integration
- Session management

- Role-based access

2. Profile System

- Personal progress tracking
- Achievement badges
- Performance statistics
- Custom settings

Leaderboard System

1. Real-time Rankings

- Global leaderboard
- Topic-specific rankings
- Weekly/Monthly challenges
- Achievement showcase

Database Design

Core Models

1. Player Model

```
model Player {
  id      String  @id
  name    String
  email   String  @unique
  progress Progress[]
  achievements Achievement[]
}
```

2. Level Model

```
model Level {
  id      Int      @id @default(autoincrement())
  difficulty Int
  questions Question[]
  completions LevelCompletion[]
}
```

3. Question Model

```
model Question {
  id      Int      @id @default(autoincrement())
  content String
  answers Answer[]
  levelId Int
  topic   Topic    @relation(fields: [topicId], references: [id])
}
```

Installation Guide

Local Development Setup

1. Prerequisites

```
Node.js v18+  
MySQL Server  
npm or yarn
```

2. Environment Configuration

```
DATABASE_URL="mysql://username:password@localhost:3306/guhuza_quiz"  
NEXTAUTH_SECRET="your-secret-key"  
NEXTAUTH_URL="http://localhost:3000"
```

3. Installation Steps

```
git clone https://github.com/yourusername/GuhuzaQuiz.git  
cd GuhuzaQuiz  
npm install  
npx prisma generate  
npx prisma migrate dev  
npm run dev
```

Docker Setup

1. Using Docker Compose

```
docker-compose up -d
```

2. Environment Variables

```
DATABASE_URL=mysql://user:password@db:3306/guhuza_quiz  
NEXTAUTH_SECRET=your-secret-key  
NEXTAUTH_URL=http://localhost:3000
```

Deployment Options

Vercel Deployment

1. Connect GitHub repository
2. Configure environment variables
3. Deploy automatically

Traditional Hosting

1. Build application
2. Configure web server

3. Set up SSL certificates
4. Configure domain settings

Security Considerations

Authentication

- Secure session management
- Password hashing
- OAuth implementation
- Rate limiting

Data Protection

- Input validation
- SQL injection prevention
- XSS protection
- CSRF tokens

API Security

- Rate limiting
- Request validation
- Error handling
- Logging system

Maintenance and Updates

Regular Tasks

1. Database backups
2. Log rotation
3. Security updates
4. Performance monitoring

Monitoring

1. Error tracking
2. Performance metrics
3. User analytics
4. Server health checks