Design Document
Riddlit: Community Puzzle Platform
CS5610 - Web Development

Ganesh Umasankar Vignesh Pakkam Saravanan

1 Project Description

Riddlit is a community-driven web platform where users create and solve brain teasers, riddles, logic puzzles, math problems, and trivia questions. The platform combines social features with gamification to encourage daily engagement and foster a vibrant puzzle-solving community.

1.1 Target Audience

Riddlit is designed for puzzle enthusiasts who enjoy both creating and solving challenges. The platform serves three primary user groups:

- Puzzle Creators: Users who want to share their creativity by crafting original riddles and brain teasers
- Puzzle Solvers: Users seeking daily mental challenges and tracking their solving progress
- Community Members: Users who engage through voting, commenting, and competing on leaderboards

2 User Personas

2.1 Puzzle Creator - Alex

Alex is a 25-year-old software engineer who enjoys crafting clever riddles and logic puzzles during breaks. Alex wants to share creative puzzles with a community, see which puzzles are most popular, track how many people solve them, and build a reputation as a skilled puzzle creator. Alex expects quick puzzle creation without complex interfaces and needs instant statistics on puzzle performance.

2.2 Puzzle Solver - Jamie

Jamie is a 32-year-old teacher who uses puzzles as a morning mental warm-up routine. Jamie wants to solve at least one puzzle daily to maintain a streak, browse puzzles by difficulty to match current mood, track personal progress over time, and discuss tricky puzzles with others. Jamie expects immediate feedback on answers and clear categorization to find preferred puzzle types.

2.3 Community Enthusiast - Morgan

Morgan is a 28-year-old data analyst who enjoys competitive aspects and community interaction. Morgan wants to climb the leaderboard through consistent solving, vote on puzzle quality to help others find good content, discuss solutions in comments, and compete with friends. Morgan expects transparent leaderboard calculations and easy ways to engage with community features.

3 User Stories

3.1 Puzzle Creator Stories (Ganesh Umasankar)

- As a new user, I want to register with a unique username and password to establish my identity as a puzzle creator.
- As a puzzle creator, I want to create puzzles with a question, answer, hint, category, and difficulty level so solvers have all the information they need.
- As a creator, I want to categorize my puzzles by type (Riddle, Logic, Math, Trivia) so users can find puzzles matching their interests.
- As a creator, I want to set difficulty levels (Easy, Medium, Hard) so solvers can choose appropriate challenges.
- As a creator, I want to view all puzzles I've created in a "My Puzzles" section so I can manage my content.
- As a creator, I want to see detailed statistics on each puzzle (total attempts, successful solves, solve rate percentage) so I understand puzzle performance.
- As a creator, I want to edit my own puzzles to fix typos or improve clarity.
- As a creator, I want to delete only my own puzzles if I want to remove them.
- As a creator, I want to see how many likes and dislikes my puzzles receive to gauge community feedback.

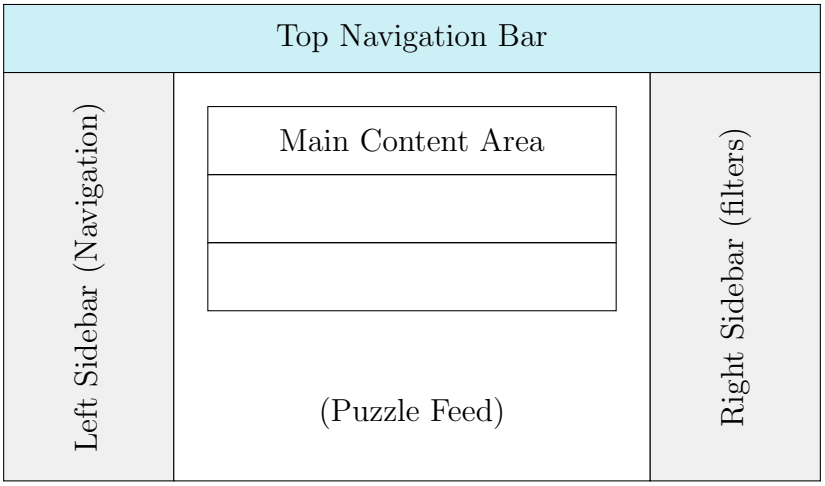
3.2 Puzzle Solver Stories (Vignesh Pakkam Saravanan)

- As a solver, I want to register with a username and password so I can track my progress.
- As a solver, I want to browse available puzzles by difficulty level or category so I can find puzzles matching my current mood.
- As a solver, I want to select a puzzle and submit my answer to test my solution.
- As a solver, I want to receive instant feedback on whether my answer is correct or incorrect so I know immediately if I solved it.
- As a community member, I want to add comments to puzzles to discuss.
- As a commenter, I want to delete my own comments if I change my mind.
- As a user, I want to vote on puzzles with likes or dislikes to show my appreciation or provide feedback.
- As a voter, I want to see community voting statistics to understand puzzle reception.
- As a solver, I want to track my solve streak (consecutive days solving at least one puzzle) so I stay motivated.
- As a solver, I want to see my personal statistics (total puzzles solved, success rate, current streak) to monitor progress.
- As a competitive user, I want to see community statistics and leaderboard to compare my performance with others.

4 Design Mockups

4.1 Layout Architecture

Riddlit uses a three-column layout inspired by Reddit's design patterns:



4.2 Navigation Structure

Top Navbar: Logo, dark mode toggle, user greeting, login/logout/register buttons
Left Sidebar: Home, category filters (Riddle, Logic, Math, Trivia), My Puzzles, My Stats, Leaderboard
Right Sidebar: Create Puzzle, Difficulty filters (Easy, Medium, Hard)

4.3 Puzzle Card Design

Each puzzle displays as a card with header (username, difficulty badge), question area, action buttons (vote, comment, solve), and statistics (attempts, success rate). Comments appear below in a collapsible section.

5 Technical Overview

5.1 Technology Stack

- HTML5, CSS3, Bootstrap 5.3.0, Bootstrap Icons
- Vanilla JavaScript ES6+ modules with client-side rendering
- Node.js with Express 5.1.0
- MongoDB 6.20.0 native driver (no Mongoose)
- express-session for authentication, CORS middleware
- ESLint 9.37.0, Prettier 3.6.2, Nodemon 3.1.10
- Docker for MongoDB deployment

5.2 Data Structure

Riddlit uses MongoDB with five collections: **Users** (authentication and statistics), **Puzzles** (content with full CRUD), **Submissions** (answer history with CRD), **Comments** (discussions with CRD), and **Votes** (likes/dislikes with full CRUD). All pages use client-side rendering with vanilla JavaScript fetching data from Express API endpoints.

6 Key Features

- **User Authentication**: Session-based authentication with username/password registration. Sessions persist for 24 hours with middleware protecting routes.
- **Puzzle Management**: Create puzzles with question, answer, hint, category, and difficulty. View, edit, and delete own puzzles with statistics on solve rates.
- **Instant Answer Validation**: Normalized answer comparison with immediate visual feedback. Automatic statistics updates for attempts and success rates.
- **Community Voting**: Like/dislike system with one vote per user per puzzle. Real-time vote counts with visual indicators for user's existing votes.
- **Comment System**: Add and delete comments on puzzles. Comments display with author username and timestamp in chronological order.
- **Personal Statistics**: Track total puzzles solved, success rate, and current solving streak. Streak counts consecutive days with correct solves.
- **Community Leaderboard**: Rankings showing top solvers by correct submissions and top creators by puzzle popularity. Paginated results with real-time updates.
- **Dark Mode**: Toggle between light and dark themes with LocalStorage persistence. Smooth transitions with proper contrast for all components.

7 Implementation Challenges & Solutions

- **Session Management**: Maintaining authentication state across multiple HTML pages without SPA framework. Created `auth.js` module checking session on every page load with backend middleware validation.
- **Real-time Statistics**: Keeping puzzle statistics accurate with simultaneous user interactions. Used MongoDB atomic operations to prevent race conditions and cached frequently accessed stats.
- **Answer Validation**: Handling different user input formats (spaces, capitalization, punctuation). Normalized both user input and correct answer to lowercase and trimmed whitespace.
- **Dark Mode Text Visibility**: Ensuring readability in both themes, especially Bootstrap components. Created comprehensive `dark-mode.css` with targeted selectors and CSS custom properties.
- **Client-Side Rendering Performance**: Rendering many puzzle cards efficiently without frameworks. Implemented pagination, used document fragments for batch DOM insertions, and event delegation for interactions.

8 What Makes Riddlit Special

- **Reddit-Inspired Interface**: Three-column layout with left navigation, center feed, and right statistics creates a familiar and intuitive user experience
- **Instant Gratification**: Immediate feedback on puzzle submissions creates an addictive loop of challenge and reward
- **Gamification Elements**: Streaks, leaderboards, and statistics transform puzzle-solving into an engaging competitive experience
- **Clean Architecture**: Modular code organization with clear separation of concerns makes the codebase maintainable and extensible
- **Modern JavaScript Practices**: Consistent use of ES6 modules, async/await patterns, and modern DOM APIs demonstrates current best practices
- **Responsive Design**: Bootstrap grid system combined with custom CSS creates seamless experience across all device sizes
- **Performance Optimization**: Efficient database queries, pagination, and event delegation ensure smooth performance with large datasets
- **Social Features**: Voting and commenting transform Riddlit from a simple puzzle site into a social platform where users interact
- **Competitive Elements**: Leaderboards and streaks encourage daily engagement and foster friendly competition