HANGMAN GAME PROJECT REPORT

OVERVIEW

This report documents the development and features of a web-based Hangman game implemented using HTML, CSS, and JavaScript. The game offers an interactive experience where players guess letters to reveal a hidden word before making too many incorrect attempts.

Hangman is a word-guessing game where players try to reveal a hidden word by picking letters. Each wrong guess adds a part to a stick-figure drawing (the "hangman"). You win by guessing the word before the hangman is fully drawn, or lose if the drawing completes first.

Technical Specifications

- **Platform**: Web browser (client-side only)
- **Technologies Used**:
    - HTML5 for structure
    - CSS3 for styling and responsive design
    - JavaScript for game logic and interactivity
- **Compatibility**: Works on all modern browsers
- **Responsive Design**: Adapts to different screen sizes

Game Features

Core Gameplay

- Random word selection from three difficulty levels
- Visual hangman drawing that progresses with wrong guesses
- On-screen keyboard for letter selection
- Word display with blanks for unguessed letters
- Maximum of 6 wrong guesses before game over

Difficulty Levels

1. **Easy**: 4-5 letter common words
2. **Medium**: 8-10 letter words
3. **Hard**: 11+ letter challenging words

User Interface Elements

- Clean, modern design with responsive layout
- Visual hangman drawing that builds with each wrong guess
- Color-coded keyboard (green for correct, red for wrong)
- Game status messages
- Hint system (reveals one letter)
- New game/reset functionality

Technical Implementation

Key Components

1. **Game State Management**:
    - Tracks guessed letters
    - Counts wrong attempts
    - Manages game over conditions
2. **Dynamic Rendering**:
    - Keyboard generated programmatically
    - Word display updates in real-time
    - Hangman drawing built progressively
3. **Event Handling**:
    - Letter button clicks
    - Difficulty selection
    - Game reset
    - Hint requests

Code Structure

- **HTML**: Defines game container and UI elements
- **CSS**: Handles styling and responsive behavior
- **JavaScript**:
    - Word banks for each difficulty
    - Hangman drawing functions
    - Game initialization and control

- User interaction handlers

Development Challenges

1. **Dynamic Hangman Drawing**:
    - Implemented as a series of functions that render individual parts
    - Called sequentially based on wrong guess count
2. **Keyboard State Management**:
    - Visual feedback for used/correct/wrong letters
    - Disabling buttons appropriately
3. **Game Flow Control**:
    - Preventing further input after game over
    - Proper reset functionality

Conclusion

This Hangman game implementation provides a complete, functional experience with multiple difficulty levels and a clean user interface. The client-side architecture makes it easy to deploy without server requirements, while the modular JavaScript code allows for straightforward expansion and modification.

The project demonstrates effective use of core web technologies to create an interactive game with proper state management and user feedback. It serves as a solid foundation that could be extended with additional features and polish.