**PROJECT INTRODUCTION:**

The Hangman game project is a classic word-guessing game implemented using HTML, CSS, and JavaScript. The game provides an interactive interface where players can choose a category, guess letters, and attempt to reveal a hidden word. The visual representation of the game includes a canvas for drawing the hangman figure, dynamic letter buttons, and a responsive design for an optimal user experience.

**PROBLEM STATEMENT:**

The project addresses the challenge of creating an engaging and visually appealing Hangman game within a web browser. The main problems to be addressed include:

**User Interaction:** Develop an intuitive and user-friendly interface for selecting game options and inputting letter guesses.

**Word Generation:** Implement a mechanism to randomly select words from predefined categories (fruits, animals, countries) and display them as a series of underscores.

**Game Logic:** Implement the core game logic, tracking correct and incorrect letter guesses, updating the displayed word, and managing the state of the hangman figure.

**Visual Representation:** Create a visually appealing representation of the hangman figure using an HTML canvas and update it based on the number of incorrect guesses.

**Game Outcome Display:** Display the result of the game (win or lose) in a popup, including the correct word in case of a loss.

**WORKING OF CODE:**

**HTML:**

* The HTML file sets up the basic structure of the webpage.
* It includes a head section with meta tags and links to external stylesheets and scripts.
* The body contains a container div that holds various elements such as options, letters, canvas, and a new game popup.

**CSS:**

* The CSS file defines the styling for different elements of the game.
* It sets up the overall styling, fonts, and background color for the webpage.
* Styles for buttons, letter containers, canvas, and different messages (win or lose) are specified.
* The game layout is designed to be visually appealing with a responsive design.

**JAVASCRIPT:**

* The JavaScript file contains the game logic and functionality.
* It initializes variables to reference HTML elements like containers, buttons, and canvas.
* An object named options holds word categories (fruits, animals, countries).
* Functions are created for displaying game options, generating a word, initializing the game, and drawing the hangman figure on the canvas.
* Event listeners are set up for letter buttons, the new game button, and window load.
* The game logic tracks correct and incorrect guesses, updates the displayed word, and manages the hangman figure based on incorrect guesses.

**GAME FLOW:**

* On window load, the game is initialized, displaying category buttons and setting up the canvas.
* When a category button is clicked, a word is randomly chosen from the selected category, and underscores are displayed to represent the word.
* The player clicks on letter buttons to make guesses. The game updates the displayed word and the hangman figure accordingly.
* If the player successfully guesses the word, a win message is displayed. If the hangman figure is completed (6 incorrect guesses), a lose message is shown.
* The new game button resets the game for another round.

**CANVAS DRAWING:**

* The canvas is used to draw the hangman figure.
* The canvasCreator function sets up a drawing context and provides functions to draw different parts of the hangman (head, body, arms, and legs).
* The drawMan function uses this canvas functionality to draw the hangman based on the number of incorrect guesses.

**CONCLUSION:**

In summary, the Hangman game works by combining HTML for structure, CSS for styling, and JavaScript for game logic and interactivity. The user interacts with the game by selecting categories and guessing letters, and the game responds by updating the displayed word and hangman figure. The canvas is dynamically drawn to represent the hangman's state. The result of the game is displayed in a popup, and players can start a new game for another round.