Mafia Game Requirements

# Epic 1: Game Setup & Configuration

* **Description:** Covers all aspects of initializing a new game, including player management, role selection, and basic game settings.
* **Functional Group(s):** Database Access, Players, Roles

## User Story 1.1: Player Management - Creation

* **As a Game Moderator, I want to** be able to add new players to the game by providing their name and an optional avatar/image, **so that** they can participate in the game.
* **Acceptance Criteria:**
* System allows input for player name.
* System allows uploading or selecting a player avatar (e.g., from Supabase bucket).
* Player data (name, avatar URL, unique ID, initial alive status) is saved to the database.
* Feedback is provided on successful player creation.
* *(Files: add-player.html, player-selection.js, database.js)*

## User Story 1.2: Player Management - Editing

* **As a Game Moderator, I want to** be able to edit existing player details (name, avatar), **so that** I can correct mistakes or update player information.
* **Acceptance Criteria:**
* System lists existing players for selection.
* System allows modification of selected player's name and avatar.
* Updated player data is saved to the database.
* *(Files: edit-player.html, player-selection.js, database.js)*

## User Story 1.3: Player Management - Deletion

* **As a Game Moderator, I want to** be able to remove players from the game roster, **so that** only intended participants are included.
* **Acceptance Criteria:**
* System allows selection of players for deletion.
* Confirmation is required before deletion.
* Player data is removed from the database or marked as inactive.
* *(Files: player-selection.js, database.js)*

## User Story 1.4: Player Selection for Game

* **As a Game Moderator, I want to** select a specific set of created players to participate in the current game instance, **so that** not all players in the database have to play every game.
* **Acceptance Criteria:**
* Interface to view all available players.
* Mechanism to select/deselect players for the upcoming game.
* The selected list of players is stored as part of the game state.
* *(Files: player-selection.html, player-selection.js, game.js, database.js)*

## User Story 1.5: Role Selection & Customization

* **As a Game Moderator, I want to** select the roles that will be included in the game from a predefined list of roles, **so that** I can customize the game's complexity and balance.
* **Acceptance Criteria:**
* System displays available roles with their descriptions and team affiliations (Mafia, Civilian, Neutral).
* Moderator can specify the number of each role to be included, ensuring the total number of roles matches the number of selected players.
* The selected roles and their counts are stored for the current game instance.
* *(Files: role-selection.html, role-selection.js, roles.js, role-customize.html, role-customize.js, game.js)*

## User Story 1.6: Automatic Role Assignment

* **As the System, I want to** randomly assign the selected roles to the selected players once the game starts, **so that** role distribution is fair and secret.
* **Acceptance Criteria:**
* Each selected player is assigned one role from the chosen set of roles.
* Role assignments are stored securely in the game state, linked to player IDs.
* The assignment process ensures the correct number of each role is distributed.
* *(Files: game.js, database.js)*

## User Story 1.7: Game Initialization

* **As a Game Moderator, I want to** start the game after players and roles are configured, **so that** the game flow can begin.
* **Acceptance Criteria:**
* A "Start Game" action triggers the initialization of the game state (e.g., setting current round to 0, game phase to Introduction).
* Initial game state is saved to the database.
* System navigates to the first phase of the game (Introduction Day).
* *(Files: player-selection.js or game.js, database.js, index.html leading to intro-day.html)*

# Epic 2: Game Flow Management

* **Description:** Defines the different phases of the game and the transitions between them.
* **Functional Group(s):** Sequences, Day Action, Night Action

## User Story 2.1: Introduction Day Phase (Round 0)

* **As a Player, I want to** participate in an initial introduction round where no actions occur but players can get acquainted, **so that** the game begins smoothly.
* **Acceptance Criteria:**
* Displays "Introduction Day - Round 0".
* Shows list of all alive players.
* No voting or special actions are enabled.
* A "Next Phase" button transitions the game to the First Identification Night.
* isIntroductionDay flag is true.
* *(Files: intro-day.html, intro-day.js, day-phase.js (handles flags))*

## User Story 2.2: First Identification Night Phase

* **As a Player, I want to** be shown my assigned role and (if applicable) my teammates, **so that** I understand my objectives and alliances.
* **Acceptance Criteria:**
* Displays "First Identification Night".
* Each player sees their own role card.
* Mafia members see a list of their teammates.
* Other roles see relevant information according to their abilities (e.g., Professional sees their status).
* No eliminations or game-altering actions occur in this phase.
* A "Next Phase" button transitions the game to Day Phase Round 1.
* hadFirstIdentificationNight flag is set to true.
* currentRound is incremented to 1.
* *(Files: role-identification.html, role-identification.js, game.js, roles.js)*

## User Story 2.3: Day Phase

* **As a Player, I want to** participate in a day phase to discuss events, make accusations, and eventually vote, **so that** we can identify and eliminate perceived threats.
* **Acceptance Criteria:**
* Displays "Day Phase - Round X".
* Shows list of alive players.
* May display information about who was eliminated during the previous night.
* Players can engage in discussion (facilitated by a timer).
* A "Next Phase" button becomes active (e.g., after discussion timer ends or by moderator action).
* Button text indicates the next phase (e.g., "Next Phase (Voting)" or "Proceed to Bomb Defusing" if a bomb is active).
* Transitions to Voting Phase or Bomb Defusing Phase.
* *(Files: day-phase.html, day-phase.js, timer.js)*

## User Story 2.4: Voting Phase

* **As a Player, I want to** be able to vote for a player to be eliminated during the day, **so that** we can remove suspected Mafia members or other threats.
* **Acceptance Criteria:**
* Displays "Voting Phase - Round X".
* Lists all alive players as potential voting targets.
* Each alive player can cast one vote.
* System records votes.
* After voting, the player with the most votes is put on "defense."
* Transitions to Voting Defense Phase.
* *(Files: voting-phase.html, voting-phase.js)*

## User Story 2.5: Voting Defense Phase

* **As the Accused Player, I want to** have a chance to defend myself before a final elimination vote, **so that** I can try to convince others of my innocence.
* **As Other Players, I want to** hear the accused player's defense, **so that** I can make an informed final vote.
* **Acceptance Criteria:**
* Displays the player on defense.
* Accused player may have a timed period to speak.
* Other players can then vote to "Execute" or "Spare" the accused.
* If "Execute" votes are a majority, the player is marked as eliminated.
* Player's role might be revealed upon elimination (configurable).
* System checks for win conditions.
* If no win condition, transitions to Night Phase.
* *(Files: voting-phase-defense.html, voting-phase.js)*

## User Story 2.6: Night Phase

* **As a Player with a night action (e.g., Mafia, Professional, Bomber), I want to** be able to perform my role-specific action secretly, **so that** I can work towards my team's objectives.
* **Acceptance Criteria:**
* Displays "Night Phase - Round X".
* Each player with a night action sees their specific action interface.
* (Sub-stories for each role action below)
* Actions are recorded by the system.
* After all actions (or a timer), the phase ends.
* System processes night actions (e.g., eliminations, bomb plants).
* System checks for win conditions.
* If no win condition, transitions to Day Phase of the next round. currentRound is incremented.
* *(Files: night-phase.html, night-phase.js, roles.js)*

## User Story 2.7: Game State Persistence

* **As the System, I want to** persist the game state (current round, phase, player statuses, votes, actions) in localStorage and synchronize with a backend database (Supabase), **so that** the game can be resilient to browser refreshes and potentially allow for reconnection or observation.
* **Acceptance Criteria:**
* Game state variables are consistently updated in localStorage.
* Key game state changes are pushed to Supabase.
* On page load for any game phase, the system attempts to load the state from localStorage and then potentially verifies/updates from Supabase.
* *(Files: database.js, supabase-init.js, all phase-specific JS files)*

# Epic 3: Role-Specific Abilities & Actions

* **Description:** Details the unique actions and characteristics of each role in the game.
* **Functional Group(s):** Roles, Night Action, Day Action, Database Access

## User Story 3.1: Mafia - Night Kill

* **As a Mafia member, I want to** collectively (or by a designated leader) choose a target to eliminate during the night phase, **so that** we can reduce the number of civilians.
* **Acceptance Criteria:**
* Mafia members see a list of alive non-Mafia players.
* They can discuss and select one target.
* The selected target is marked for elimination by the Mafia.
* Only one Mafia kill action per night.
* *(Files: night-phase.html, night-phase.js, roles.js)*

## User Story 3.2: Professional - Night Shot

* **As a Professional, I want to** choose a player to shoot during the night phase, **so that** I can try to eliminate a Mafia member.
* **Acceptance Criteria:**
* Professional sees a list of all other alive players.
* Professional selects one target to shoot.
* If the target is Mafia (Regular Mafia, Don, etc.), the target is eliminated. The Professional may or may not know the outcome immediately.
* If the target is a Civilian, the Professional is also eliminated (or suffers a penalty like losing their ability). (This needs to be clarified from code - current summary suggests this is a check).
* Only one shot per night (or limited shots per game).
* *(Files: night-phase.html, night-phase.js, roles.js)*

## User Story 3.3: Bomber - Plant Bomb

* **As a Bomber, I want to** choose a player to plant a bomb on during the night phase, **so that** I can create a threat that needs to be addressed by the civilians.
* **Acceptance Criteria:**
* Bomber sees a list of alive players.
* Bomber selects one target.
* A "bomb" (with a unique defusal code) is associated with the target player.
* The fact that a bomb is planted (but not necessarily on whom, or the code) is announced at the start of the next Day Phase.
* The bomb might have a timer or trigger condition for detonation if not defused.
* *(Files: night-phase.html, night-phase.js, roles.js)*

## User Story 3.4: Bomb Defusal Phase

* **As a Civilian Player, I want to** attempt to defuse a bomb planted on a player, **so that** we can prevent an elimination and a Bomber victory.
* **Acceptance Criteria:**
* If a bomb is active, the Day Phase transitions to Bomb Defusal.
* Players are presented with an interface to guess the defusal code.
* A limited number of attempts or a time limit for defusal.
* If the correct code is entered, the bomb is defused. The Bomber might be revealed or eliminated.
* If the bomb is not defused in time (or attempts run out), the player with the bomb is eliminated. This might also trigger a win for the Bomber if it's their win condition.
* After defusal/detonation, game proceeds (e.g., to voting or next night).
* *(Files: bomb-defuse.html, bomb-defuse.js, day-phase.js)*
* **(Add User Stories for other roles like Doctor, Detective, Don, etc., if their specific night/day actions are implemented. Analysis of roles.js and night-phase.js would be key here.)**
* Example: **User Story 3.X: Doctor - Night Heal**
* **As a Doctor, I want to** choose a player (including myself) to protect during the night, **so that** they cannot be eliminated by the Mafia.
* **Acceptance Criteria:**
* Doctor selects one target.
* If the target is chosen for elimination by Mafia, the elimination is prevented.
* Doctor cannot heal the same person two nights in a row (common rule, check implementation).

# Epic 4: Game State Display & UI/UX

* **Description:** Focuses on how game information is presented to the players.
* **Functional Group(s):** All (as UI is pervasive)

## User Story 4.1: Player Status Display

* **As a Player, I want to** clearly see the list of all players and their current status (e.g., Alive, Eliminated), **so that** I can track who is still in the game.
* **Acceptance Criteria:**
* Player avatars and names are displayed.
* Status is visually distinct (e.g., grayscale for eliminated players).
* Player sequence/order is maintained if applicable.
* *(Files: day-phase.html, day-phase.js, night-phase.html, night-phase.js, character-cards.css)*

## User Story 4.2: Role Card Display

* **As a Player, I want to** be able to view my own role card with its description, abilities, and team affiliation, **so that** I always know my objectives.
* **Acceptance Criteria:**
* Accessible during relevant phases (e.g., Identification Night, own turn at Night).
* Displays role name, image, description, win condition.
* *(Files: role-identification.html, role-identification.js, view-roles.html, view-roles.js, character-cards.css)*

## User Story 4.3: Game Phase and Round Display

* **As a Player, I want to** clearly see the current game phase (e.g., Day, Night, Voting) and the current round number, **so that** I understand the game's progression.
* **Acceptance Criteria:**
* Displayed prominently on all game screens.
* *(Files: All phase-specific HTML/JS files, e.g., day-phase.js function updateGameStatus)*

## User Story 4.4: Timers

* **As a Player, I want to** see visual timers for timed phases (e.g., day phase discussion, defense speech), **so that** I know how much time is remaining.
* **Acceptance Criteria:**
* Timer countdown is displayed.
* Actions may be triggered or disabled when the timer expires.
* *(Files: timer.js, timer.html, timer.css, integrated into phase-specific JS)*

## User Story 4.5: Responsive Design & Styling

* **As a Player, I want** the game to be visually appealing and usable across different screen sizes (desktop, potentially tablet), **so that** I have a good user experience.
* **Acceptance Criteria:**
* Consistent styling (fonts, colors, themes like dark theme) is applied.
* Layout adapts reasonably to different viewport sizes.
* Game elements (buttons, cards) are easily interactive.
* *(Files: styles.css, dark-theme.css, character-cards.css, all HTML files for structure)*

## User Story 4.6: Game Event Notifications/Log

* **As a Player, I want to** be informed about key game events (e.g., who was eliminated, results of votes, bomb planted), **so that** I can make informed decisions.
* **Acceptance Criteria:**
* Clear messages are displayed at appropriate times (e.g., start of Day phase).
* A game log might be available to review past events (check if implemented).
* *(Files: Phase-specific JS files updating the DOM)*

# Epic 5: Winning & Losing Conditions

* **Description:** Defines how the game ends and how winners are determined.
* **Functional Group(s):** Winning Criteria, Database Access

## User Story 5.1: Mafia Win Condition

* **As the System, I want to** declare the Mafia team as winners when the number of Mafia members is equal to or greater than the number of Civilian team members, **so that** the game ends appropriately.
* **Acceptance Criteria:**
* After each elimination or phase change, the system checks this condition.
* If met, the game ends, and Mafia victory is announced.
* *(Files: game.js, day-phase.js, night-phase.js - wherever eliminations are processed)*

## User Story 5.2: Civilian Win Condition

* **As the System, I want to** declare the Civilian team as winners when all Mafia members (and potentially other hostile neutral roles) have been eliminated, **so that** the game ends appropriately.
* **Acceptance Criteria:**
* After each elimination, the system checks if any Mafia players remain.
* If no Mafia remain, the game ends, and Civilian victory is announced.
* *(Files: game.js, day-phase.js, night-phase.js)*

## User Story 5.3: Bomber Win Condition (Example - verify from code)

* **As the System, I want to** declare the Bomber as a winner if their planted bomb successfully detonates on a target, OR if they are the last player alive, OR if they meet specific unique conditions for their role, **so that** neutral roles can achieve victory.
* **Acceptance Criteria:**
* Check if bomb detonation (eliminating the target) is a standalone win.
* Check if specific conditions for Bomber victory are implemented.
* *(Files: bomb-defuse.js, game.js, roles.js)*
* **(Add User Stories for other Neutral role win conditions if applicable.)**

## User Story 5.4: Game Over Screen

* **As a Player, I want to** see a game over screen when a win condition is met, displaying the winning team/player and potentially revealing all remaining roles, **so that** the game concludes clearly.
* **Acceptance Criteria:**
* Displays winning faction and reason for victory.
* Optionally shows a summary of roles of all players.
* Provides options to play again or return to the main menu.
* *(Files: game-over.html, summary.html, summary.js)*

# Epic 6: Database & Backend Integration

* **Description:** Covers the interaction with the Supabase backend for data persistence and real-time features.
* **Functional Group(s):** Database Access

## User Story 6.1: Supabase Initialization

* **As the System, I want to** initialize a connection to the Supabase backend using predefined API keys and URLs, **so that** data can be stored and retrieved.
* **Acceptance Criteria:**
* Supabase client is created successfully.
* Connection status can be monitored (e.g., for diagnostics).
* *(Files: supabase-init.js, config.js, connection-status.js)*

## User Story 6.2: Storing Player Data

* **As the System, I want to** store player information (ID, name, avatar, role, status) in Supabase tables, **so that** player data is persistent and centrally managed.
* **Acceptance Criteria:**
* CRUD operations (Create, Read, Update, Delete) for player data are functional against Supabase.
* *(Files: database.js, player-selection.js)*

## User Story 6.3: Storing Game State

* **As the System, I want to** store the overall game state (current game ID, active players, assigned roles, current round, current phase, votes, actions) in Supabase, **so that** game progression is saved and can potentially be restored or observed.
* **Acceptance Criteria:**
* Game state is updated in Supabase at critical points (e.g., phase transitions, actions taken).
* Ability to fetch the current game state from Supabase.
* *(Files: database.js, all phase-specific JS files, game.js)*

## User Story 6.4: Image/Avatar Storage

* **As the System, I want to** utilize Supabase Storage for player avatars, **so that** images are hosted and managed efficiently.
* **Acceptance Criteria:**
* Avatars can be uploaded to a Supabase bucket.
* Avatar URLs are retrieved and displayed in the game.
* *(Files: database.js, player-selection.js, get-supabase-buckets.ps1 script indicates usage)*