Bullshit Multiplayer Card Game

Milestone 3 – API & Socket.io Design

Game Overview

- • Multiplayer card game: "Bullshit"
- • Goal: be the first player to get rid of all cards
- • Players claim they are playing the required rank in sequence
- • Other players can call "bullshit" if they think a player is lying
- • Server enforces all rules and tracks true vs claimed plays

High-Level Architecture

- • RESTful API built with Express
- • Socket.io for real-time communication
- • Central resources: Games, Game_Participants, Player_Hands,
- Game_Actions, Pile_Cards, Chat_Messages
- • Server is the single source of truth to prevent cheating
- • REST for full state fetch, Socket.io for live updates

Core Data Model

- • Games: overall session state (status, current_turn, current_rank,
- direction, pile count, winner)
- • Game_Participants: players in a game, turn_order, cards_in_hand
- • Player_Hands: actual cards owned by each participant
- • Game_Actions: plays, passes, challenges, pickups
- • Pile_Cards: cards currently in the middle pile

Lobby & Game Management Endpoints

- • POST /api/games – create a new game lobby
- – Sets status='waiting', creator added as first participant
- • POST /api/games/:game_id/join – join a waiting game
- – Validates capacity, status, duplicate joins
- • POST /api/games/:game_id/leave – leave a waiting game
- – Removes participant; deletes game if empty

Starting the Game

- • POST /api/games/:game_id/start
- – Only participants can start
- – Requires at least 2 players, status='waiting'
- • On start:
- – Set status='in_progress', initialize current_rank and direction
- – Shuffle 52-card deck and deal to players
- – Fill Player_Hands and update cards_in_hand
- – Set current_turn to player with lowest

Play Cards – Core Gameplay

- • POST /api/games/:game_id/play
- – Only current_turn player can call
- – 1–4 cards, claimed_rank must match games.current_rank
- – Server verifies cards exist in player hand
- • On success:
- – Move cards from Player_Hands to Pile_Cards
- – Record Game_Actions entry with claimed_rank and truthfulness

Challenge – "Bullshit" Logic

- • POST /api/games/:game_id/challenge
- – Any other participant can call within challenge window
- – Uses last Game_Actions play to evaluate truth
- • If challenge succeeds (play was a lie):
- – Last player picks up entire pile
- • If challenge fails (play was honest):
- – Challenger picks up entire pile
- • Server:

Game State & Chat Endpoints

- • GET /api/games/:game_id
- – Returns game status, current_turn, current_rank, pile_count
- – Participant list with cards_in_hand (not card details)
- – Requesting player's full hand
- – Recent actions and challenge window info
- • Chat:
- – POST /api/games/:game_id/chat – send message

Real-Time Events with Socket.io

- • Lobby events:
- – join_game / player_joined
- – leave_game / player_left
- – client:request_state / game:state
- • Gameplay events:
- – play_card / card_played
- – bullshit_option, bullshit_called, bullshit_result
- – turn_ended, game:finished

Authentication & Sessions

- • Express uses session middleware for HTTP routes
- • Socket.io shares the same session:
- – io.use attaches session to socket.request
- • authSocket middleware:
- – Checks socket.request.session.user
- – Rejects unauthenticated connections
- – Attaches user data to socket.user for events
- • Result: unified auth for both REST API and real-time layer

Design Highlights & Future Work

- • Highlights:
- – All critical game logic enforced on the backend
- – REST + Socket.io combo for reliable, real-time gameplay
- – Clear separation of concerns between resources
- – Data model supports auditing via Game_Actions
- • Possible extensions: