# PRD: THEE BOGGLE BOOST!

Status: Draft | Current | Needs Update | Obsolete

## Background:

THEE BOGGLE BOOST is a community-driven, web-based extension of the classic Boggle word game. Instead of playing alone, users can create, share, and complete custom Boggle-style puzzles. Each challenge contains a letter grid, clues, and correct answers. The app introduces social features such as daily global challenges, real-time leaderboards, hints, accessibility options, and personalized challenge links.

The purpose of this product is to build an interactive learning and gaming platform where vocabulary development, competition, and creativity merge. The system uses React (frontend), Django (backend), and Firebase (authentication + real-time data), making it scalable and accessible across devices.

- 1-Pager: Boggle boost pitch
- UX Mocks: boogle_boost_wireframe
- Design link : Boggle Boost Design doc

# Requirements

## Functional Requirements

**FR-01** – Log in / log out (Core)

- As a user, I will be able to log in to Boggle Boost using my Google account.
- As a user, I will be able to log out when I am done.
- When I am logged in, I will be treated as a **registered** user; when I am not logged in, I will be treated as a **guest**.

**FR-02** – Create challenges with multiple clues and answers (Core)

- As a registered user, I will be able to open a **Create Challenge** screen.
- I will be able to:
  - Choose a **difficulty level** (e.g., Easy, Medium, Hard).
  - Choose a **grid size** (4×4, 5×5, 6×6).
  - Configure the letters on the grid (including any special tiles if supported).
  - Optionally add **clues** or a short description.

- I will be able to click a **"Generate Dictionary"** button to build a list of valid words for the current grid.
- When I save the challenge, the game will store the grid, dictionary, difficulty, clues, and my identity as creator, so the challenge has **many correct answers**.

**FR-03** – View challenges I have created (Core)

- As a registered user, I will be able to open a **"My Challenges"** page.
- I will see a list of all challenges I have created, including at least:
    - Title/label
    - Creation date
    - Intended recipient(s)
    - Status (e.g., active / deleted).

**FR-04** – Delete any challenge I have created (Core)

- As a registered user, I will be able to select a challenge from **My Challenges** and delete it.
- Once deleted, that challenge will no longer be available for others to play.

**FR-05** – Play challenges created by others (Core)

- As a registered user, I will be able to open a challenge link (for example, from email or a shared link) and see the challenge board.
- I will be able to start the challenge and attempt the puzzle within a **time limit**.
- I will be able to submit answers and receive **instant feedback** on each submission.

**FR-06** – Timed play and one-word submission (Core)

- When I start a challenge or practice game, a **timer** will start based on the chosen difficulty.
- I will submit **one word at a time** through an input box.
- I will be able to submit words until the timer reaches zero or I end the game early.
- When the timer reaches zero, the game will stop accepting new answers and show a summary/results screen.

**FR-07** – Dictionary-based correctness and case-insensitive answers (Core)

- For each board, the game will maintain a **dictionary** of valid words tied to that board.
- When I submit a word, the game will:
    - Normalize it to a consistent form (e.g., lowercase) so input is **case-insensitive**.
    - Check whether it is in the board's dictionary (and, if full Boggle rules are enforced, whether it can be formed on the grid without reusing tiles and meets the minimum length).
- If the word passes the checks, it will be marked as **correct**; otherwise, as **incorrect**.

**FR-08** – Reveal all answers and request hints (Core)

- After a challenge ends (timer expires or I give up), I will be able to see the **full list of valid words** for that board, with my found words distinguished from missed words.
- While playing, I will be able to request **hints** (for example, show the first letter, word length, or lightly highlight a starting tile), subject to reasonable limits.

**FR-09** – Play in practice / solitaire mode (Core)

- As any user (guest or registered), I will be able to start a **practice/solitaire** game.
- I will choose a difficulty and grid size, then play with the same rules (timer, word submission, feedback) as normal challenges.
- If I am a guest, my practice games do not need to be stored after the session ends.

**FR-10** – Guest vs registered user capabilities (Core)

- If I am **not logged in** (guest), I will:
  - Be allowed to play **practice/solitaire** games.
  - **Not** be allowed to create challenges, view challenge history, or complete challenges created by others.
- If I am **logged in**, I will:
  - Be able to create, view, delete, and play challenges.
  - Also be able to play practice/solitaire games.

**FR-11** – Share challenges via personalized links (Core)

- After I create a challenge, I will be able to generate a **unique link** for it.
- I will be able to share this link with others (for example via email or messaging).
- When a recipient opens the link and logs in, they will be taken directly to the challenge play screen for that puzzle.

**FR-12** – Shuffle and rotate the board (Core)

- While viewing a board, I will be able to:
  - **Shuffle** the letters to get a new arrangement, and/or
  - **Rotate** the grid (e.g., 90°, 180°, 270°).
- These actions change only the visual arrangement, not which words are valid.

**FR-13** – Difficulty levels and grids (Core)

- When creating or starting a game, I will be able to choose a **difficulty level**.
- The app will provide, at minimum:
  - **Easy:** 4×4 grid with standard letters and a generous time limit.
  - **Medium:** 5×5 grid with a moderate time limit.
  - **Hard:** 6×6 grid that may include rarer letters (Q, Z, X) and a stricter time limit.

**FR-14** – Daily global challenge (Core)

- As a registered user, I will be able to play a **daily global challenge**, a single shared puzzle for all users on that day.
- My score for the daily challenge will be recorded separately from my custom challenges.

**FR-15** – Live leaderboard and rank notifications (Core)

- As a registered user, I will be able to view **leaderboards** (e.g., for daily challenges or overall high scores).
- When I reach certain milestones (such as moving into the top 10 or reaching Rank 1), I will see **notifications** (e.g., toast messages) celebrating those achievements.
- Leaderboard data will update in near real time.

**FR-16** – Stats dashboard (Core)

- As a registered user, I will be able to open a **stats dashboard** showing:
  - Total words found
  - Accuracy (correct vs incorrect submissions)
  - Streaks or consecutive days played
  - Best scores or other relevant metrics.

**FR-17** – Custom themes (Core)

- As a user, I will be able to choose from different **visual themes** (e.g., light, dark, high-contrast, color palettes).
- The game will apply my chosen theme while I play.

**FR-18** – Privacy controls for challenges (Core)

- As a registered user, I will be able to configure **who is allowed to send me challenges**, such as:
  - Anyone
  - Only specific emails/usernames or groups
- If someone who is not allowed tries to send me a challenge, the app will block the send and show an appropriate message.

**FR-19** – View word meanings using a dictionary API (Core)

- After a game, I will be able to click/tap any word in the results list to see its **definition**.
- The app will fetch definitions from an external **dictionary service**.

**FR-20** – Automated challenge generation (Stretch)

- The system will be able to automatically generate a complete challenge (grid + dictionary) for me when I choose a difficulty level, so I don't always have to design boards by hand.

**FR-21** – Multilingual support (Stretch)

- The system will allow me to choose a **language** (e.g., English, Spanish, French).
- When I change language, the game will use a dictionary and letter distribution appropriate for that language.

**FR-22** – Mobile responsive experience (Stretch)

- The web app will adapt its layout to fit **phones, tablets, and desktops**, so I can comfortably play on multiple device types.

## System Requirements

**SR-01** – Overall architecture

- The system shall be implemented as a **web application** with a separate frontend and backend.
- The frontend shall communicate with the backend and database via HTTP/HTTPS APIs or real-time APIs where appropriate.

**SR-02** – Frontend framework

- The frontend shall be implemented using **React.js** (or a React-compatible framework).
- The frontend shall render all game screens (login, play, create, stats) and handle user input.

**SR-03** – Backend framework

- The backend shall be implemented using **Django** (or another agreed Python web framework with equivalent capabilities).
- The backend shall expose endpoints to:
    - Create, read, update, and delete challenges and challenge results.
    - Serve daily challenges and leaderboard data.
    - Proxy or coordinate calls to external services when needed.

**SR-04** – Authentication and identity

- User authentication shall be implemented using **Firebase Auth** with Google sign-in.

- The system shall associate each authenticated user with a stable unique identifier used to store and retrieve their challenges, stats, and preferences.

**SR-05** – Data storage

- Persistent data (users, challenges, challenge attempts, daily challenges, stats, and preferences) shall be stored in a **cloud database** such as **Firebase Firestore** (or an equivalent managed store).
- The data model shall support the CRUD operations required by the functional requirements (create/view/delete challenges; record results; store settings).

**SR-06** – External dictionary service

- The system shall integrate with an external **dictionary API/service** to retrieve word definitions for the “view meaning” feature.
- If the dictionary service is unavailable, the game shall continue to function; only the definition lookup feature will be degraded.

**SR-07** – Hosting and deployment

- The frontend and backend shall be deployed to a **cloud hosting platform** such as **Firebase Hosting** and/or **Vercel** (or equivalent).
- The production deployment shall be accessible over **HTTPS**.

**SR-08** – Source control and collaboration

- The codebase shall be maintained in a **GitHub** repository (or equivalent).
- Team members shall use branches and pull requests (or an agreed workflow) for collaboration.

**SR-09** – Automated testing and CI (if supported)

- The project shall include **automated tests** for core game logic (e.g., word validation, scoring, timer behavior).
- When possible, tests shall be runnable via a continuous-integration workflow (e.g., GitHub Actions) on each push or pull request.

## Non-Functional Requirements

**NF-01** – Performance

- Main pages (login, play, create, stats) should load within a reasonable time on a normal internet connection.
- When a user submits a word, feedback (correct/incorrect) should appear quickly enough to feel **instant** to the user.

**NF-02** – Accessibility

- The UI shall support **keyboard navigation** for key actions (e.g., starting a game, submitting words, navigating dialogs).
- The UI shall provide a **high-contrast color mode** and **adjustable text sizes**.
- The UI shall be compatible with **screen readers** so that visually impaired users can understand the interface.

**NF-03** – Usability and clarity

- The interface shall clearly indicate how to:
  - Start a new challenge or daily challenge
  - Submit words and view their feedback
  - View answers, hints, stats, and settings
- Notifications and messages shall use clear, concise language.

**NF-04** – Privacy and security

- The system shall enforce each user's privacy settings for who may send them challenges.
- A user's stats, history, and preferences shall not be visible to other users unless explicitly allowed (e.g., through future social features).
- Data associated with a user's account shall be stored and transmitted securely (e.g., via HTTPS and authenticated access to the database).

**NF-05** – Maintainability and collaboration

- The code shall be structured into clear modules (UI, game logic, data access, integration) to support collaborative development and testing.
- Documentation (e.g., README, setup instructions) shall be maintained so new contributors can set up and work on the project.