Games Sonnent Event Table

EVENT NUMBER	DESCRIPTION	TRIGGER	REFLECTED STATUS	SYSTEM RESPONSE
1	New User Registration	User fills and submits the "Sign Up" form on the page.	The user's account is created and stored in the database	The system validates the input, hashes the password, creates a new User record, and logs the user in by creating a session.
2	User/Admin Login	A registered user or admin submits the "Sign In" form with their email and password.	The user is authenticated, stored in the browser's session storage.	The system validates credentials against the database. On success, it generates a JSON Web Token (JWT) and redirects the user to the main page
3	Browse Trending Games	User opens homepage.	Trending games list is displayed from **RAWG API**.	Frontend sends request to RAWG API, backend fetches trending game list, returns results to UI.
4	Browse Suggested Games	User opens homepage.	Suggested games list (from MongoDB) is displayed under "Suggested Games".	Backend fetches games from **Games collection** and returns to frontend.
5	Add Game	Logged-in user submits game name in **Add Yours form**.	Game stored in **PendingGames collection** for admin approval.	Backend queries RAWG API. If game found → save in **PendingGames** with status `pending`. If not found → enable **Manual Submission Form**.
6	Manual Game Submission	RAWG API returns no result → user fills manual form with slug, name, description, image	Submitted game stored in **PendingGames collection** with status `pending`	System saves the data with all user-entered fields and user receives a toast message

		URL, genres, platforms, rating, release date, website.		“Game "${gameData.name}" submitted for admin approval! ”
7	Admin Approves game	Admin visits **/admin page**, selects a pending game, clicks Approve.	Game is added **Games collection**.And status attribute is removed	Backend saves approved game in Games collection and updates status to `approved in PendingGames collection`
8	Admin Rejects Game	Admin visits **/admin page**, clicks Reject.	Game remains in **PendingGames** with status = `rejected`.	System updates status field and saves optional `adminNotes`.
9	Admin Edits Game	Admin selects pending game and edits details.	Game record updated in DB.	System updates allowed fields (name, desc, genres, platforms, etc.) and saves changes.
10	View Game Details	User clicks on a game card (Trending/Suggested).	A **Big Card** opens showing description, genres, platforms, website link, rating, reviews.	If game exists in RAWG → fetch data from RAWG. If not → fetch from **MongoDB**.
11	Write Review	Logged-in user submits review (rating + comment) on a game.	Review saved in **Reviews collection**.	Backend validates input, saves review with username + timestamp+gameid +reviewText+rating
12	View Reviews	Any user (logged in or not) clicks on game details	Review list displayed.	System fetches reviews from **Reviews collection** sorted by latest.
13	Your Suggestions Page	Logged-in user clicks **Your Suggestions** in navbar.	Page shows games suggested by that user.	Backend queries **Games collection** where `addedBy = username`.
14	About Page	User clicks "About"	Project	Static content

		in navbar.	description and developer email shown.	rendered from frontend.
15	Logout	Logged-in user/admin clicks **Logout button**.	Session cleared, UI resets to guest state.	System removes JWT token from browser storage and shows login/register options.