GameFevr Website

Full stack website with over 18k games in database

The website provides filters to query and find games.

As a user, you can leave a review on game, rate or bookmark a game and it will be saved to their profile.

The website design was created by a professional designer, and I implemented it.

TechStack

- Next JS
- SCSS
- GSAP
- Tailwind CSS
- Mongo DB
- Type Script
- Send Grid
- Next Auth
- Zustand

Features

Reviews- Each authenticated user can leave their review on any game, and other users can see and like or dislike the reviews. The reviews are saved under the user's database collection and are visible on their profile. When posting a review, the user has the option to rank the game. If the user has already ranked the game, their previous ranking will be removed and the updated rank will be added.

Rate Game- Each authenticated user can rate any game on a scale from 1 to 4, ranging from "waste of time" to "must". The ratings are not visible to other users, but each game can display the ranking percentage based on the ratings received.

Bookmark Game- Each authenticated user can save any game, and it will be shown in their profile. They can also remove the bookmark if desired.

User Score- The website has a scoring system for each authenticated user. By performing any of the actions listed above, a score is added to the user's profile. Other users who visit the profile can see the score and even boost the user's rank.

Database Stracture

The database consists of several collections:

- games- This collection includes fully detailed game documents with fields such as website link, description, genres, platforms etc.
- games_data- This collection collects future data extracted from user activity on the website. For example, when a user ranks a game, visits it or shares it.
- short_games- This collection contains minimal information about the games, excluding the full detailed documents when they are not necessary.
- ranks- When a user ranks a game, it is saved here as a document with the game ID, user ID, the rank value, and the creation timestamp.
- reviews-When a user leaves a review on a game, it saved the document with the user ID, game ID, game name, game image, the review text, and the creation timestamp. Additionally, it creates two empty array fields, 'like' and 'dislike', for other users to express their preference by liking or disliking the review, which will be associated with their IDs.

Search And Filters Stracture

When the user applies filters from the **<Filters />** component, they are added to the URL query.
In the **getServerSideProps** function on the search page, I collect those query parameters and build a query for the MongoDB database to fetch the matching documents.
Then I return the array of game documents, the total count of matched documents for the query, and a Boolean value indicating if there is a next page for pagination.
Then the client receive the values and displays it, if there is not result, it indicates as no result found, or if there is no next page then it doesn’t show the 'Load More' button.

Optimization:

The main optimization process here was to create index for each field in the database, to reduce fetching documents time.