Version 1.0

22/09/2025

**GameRater**

Marta Grazia

uhi

Applied software development

Contents

[Project requirements 1](#_Toc209453193)

[project idea 1](#_Toc209453194)

[functional requirements 1](#_Toc209453195)

[Technical Stack 1](#_Toc209453196)

[modeling and desgin 2](#_Toc209453197)

[site map 2](#_Toc209453198)

[database entity relationship diagram 2](#_Toc209453199)

[api design routes 2](#_Toc209453200)

# Project requirements

Design a minimum viable product of a data driven application with a clean and responsive user interface.

* Front end UX and UI
* Backend API
* Data processing or integration

## project idea

For this implementation I decided to create a video game rating web application that allows the user to look for a game and add a rating. Adding a rating will automatically add the game to the user rating page where all previous ratings are saved and that can be used as a personal log of all played games. The user will be able to update their ratings as well as deleting them.

## functional requirements

* User registration
* User authentication
* Comprehensive game search via IGDB API
* Create rating for a searched game
* View previous ratings
* Update previous ratings
* Delete previous ratings
* Responsive UI
* Charts related to game preferences

## Technical Stack

Back end:

* Nodejs Express
* Sqlite database
* JWT (json web token) authentication

Front end:

* React

# modeling and desgin

## site map

A diagram of a website

AI-generated content may be incorrect.

## database entity relationship diagram

A diagram of a diagram

AI-generated content may be incorrect.

## api design routes

Authentication -> /register -> create new account

/login -> login to your account

Games -> /search -> get matching games

/rate -> add rating to game

/user-ratings -> get all user ratings

/update-rate/:id -> update rating with id

/delete/:id -> delete rating with id

AuthController -> manages authentication logic and user creation

igdbController -> manages fetching game info from IGDB API  
RatingController-> manages rating CRUD