PCMASTERRACEDB

Jacopo Foschi, Mosè Barbieri

2025/2026

Contents

1 Introduction

1.1 The website

PCMASTERRACEDB is a website that let's you browse it's catalogue of games to check their information, review them and check if you already own them or if they are in your wish-list. You can then talk about your favourite games with other users in the forums. While also managing your account using a personalised profile image.
There are then admins and the master of the site, who can both act as basic users.
Admins can and will moderate forums removing any harmful comment that any user should publish.
The master handles all game and subsequently, forum entries there are. Adding, modifying or removing them. The master can also manage users removing them.

2 Features

2.1 Users

- explore a database of games grouped by genre, or ordered by release date or rating;
- review the games you played and make your opinion count;
- share your opinions on the game forums and speak with other users;
- handle your user profile information and deletion;
- manage your personal library by labelling every game you either own or wish to own
- modify if you made a mistake your own comments or reviews or delete them

2.2 Admins

- enjoy all your features of the previous role;
- moderate forums by deleting harmful comments;
- make your status known with an admin label next to your profile name;

2.3 Superadmin

- enjoy all your features of the previous roles;
- use your control panel to see every user subscribed to the site and handle deletion;
- manage games and categories;

3 Mock-up

3.1 Figma link

Figma Link

4 Accessibility

4.1 Planning

Accessibility was a key part of design. All colors were chosen while keeping colorblind people in mind and checking that every color arrangement complies with color contrasts AAA standards.

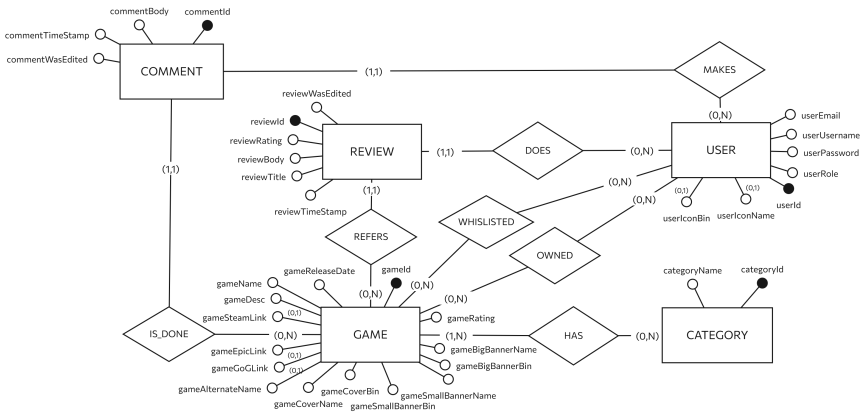
4.2 Execution

The frontend was built following the correct heading designs in order to cater to people using assistive devices and wherever possible the site is also browsable using only the jeyboard.

5 DB

5.1 E/R Schema

The backend planning started with the database. Here, you can see the E/R schema:



5.2 Seeding

The seeding is completely handled in a typescript sub-project in the back-end. It consists of a main file called seed.ts that works using promise based seeding steps that first of all creates the database and then seeds it using a mix of already executable insert into SQL queries that are read and then executed. Or done with a JSON for users and games as they also require the insertion of images.

6 Technologies used

6.1 General

- Node.js
- Express.js
- Typescript
- NPM

6.2 Backend specific

- MySQL

6.3 Front-end specific

- Vue
- Vue Router
- Vite
- Axios
- Sass

7 NPM packages

7.1 Backend

dependencies

- bcrypt
- connect-api-history-fallback
- cookie-parser
- dotenv

- express
- jsonwebtoken
- multer
- mysql2
- tsx

dev-dependencies

- @types/bcrypt
- @types/connect-history-api-fallback
- @types/cookie-parser
- @types/express
- @types/jsonwebtoken
- @types/multer
- @types/node
- typescript

7.2 Front-end

dependencies

- axios
- moment
- pinia
- vue
- vue-router

dev-dependencies

- @vitejs/plugin-vue
- sass
- sass-embedded
- typescript
- vite
- vue-tsc

8 API endpoints

8.1 Authentication

Create

- `/api/auth/register` — Creates new user

Read

- `/api/auth/profile` — Gets user profile from current session, does not retrieve complete profile
- `/api/users/exists/username-exists/:username` — Checks if username exists
- `/api/users/exists/email-exists/:email` — Checks if email exists

Update

- `/api/auth/change-password` — Changes password of currently logged-in user after validating current password

Other

- `/api/auth/login`
- `/api/auth/logout`

8.2 User

Read

- `/api/users` — Lists all users (superadmin privilege required)
- `/api/user` — Gets profile of currently logged-in user from DB, including user image

Update

- `/api/user` — Changes image and nickname of currently logged-in user

Delete

- `/api/user/:userId` — Deletes a user (self-deletion or superadmin action)

8.3 Games

Create

- `/api/games` — Creates a game (superadmin only)

Read

- `/api/games/rating` — Lists all games ordered by rating
- `/api/games/release` — Lists all games ordered by release date
- `/api/games/:genreId` — Lists all games of a given genre
- `/api/games/as-you-type/:partialName` — Lists games matching partial input
- `/api/games/matching/:partialName` — Lists all games matching search input
- `/api/game/:gameId` — Gets details of a single game
- `/api/game/all/:gameId` — Returns all details of a game

Update

- `/api/games/:gameId` — Updates game information (superadmin only)

Delete

- `/api/games/:gameId` — Deletes a game (superadmin only)

8.4 Categories

Create

- `/api/categories`

Read

- `/api/categories`

Update

- `/api/categories/:categoryId`

Delete

- `/api/categories/:categoryId`

8.5 Game Categories

Create

- `/api/game-categories`

Read

- `/api/game-categories/:gameId` — Reads all categories associated with a game

Update

- `/api/game-categories`

Delete

- `/api/game-categories/:gameId/:categoryId`

8.6 Owned

Create

- `/api/owned/:gameId`

Read

- `/api/owned` — Lists all owned games of user
- `/api/owned/:gameId` — Checks if a game is owned by the logged-in user

Delete

- `/api/owned/:gameId`

8.7 Wishlist

Create

- `/api/wishlist/:gameId`

Read

- `/api/wishlist` — Lists all wishlisted games of user

Update

- `/api/wishlist/:gameId` — Checks if a game is wishlisted by the logged-in user

Delete

- `/api/wishlist/:gameId`

8.8 Reviews

Create

- `/api/reviews/:gameId`

Read

- `/api/reviews/game/:gameId` — Lists all reviews of a game, prioritizing user's review if logged in
- `/api/reviews/user` — Lists all reviews of the user
- `/api/reviews/permission/:gameId` — Checks if user already wrote a review

Update

- `/api/reviews/:gameId` — Updates logged user's review

Delete

- `/api/reviews/:gameId` — Deletes logged user's review

8.9 Forum

Create

- `/api/games/:gameId/comments`

Read

- `/api/forums` — Lists all forums
- `/api/forums/as-you-type/:partialName` — Lists forums matching partial input
- `/api/forums/matching/:partialName` — Lists forums matching search input
- `/api/games/:gameId/comments` — Lists all comments of a game
- `/api/games/:gameId/banner` — Gets forum banner

Update

- `/api/comments/:commentId` — Updates comment

Delete

- `/api/comments/:commentId` — Deletes comment

9 Project structure

9.1 Work-tree

```
ElaboratoISW
|
+-- backend
|   |
|   +-- db-src
|   |   |
|   |   +-- assets
|   |   |   |
|   |   |   +-- bigBanners
|   |   |   +-- covers
|   |   |   +-- smallBanners
|   |   |   +-- users
|   |   |
|   |   +-- data
|   |   +-- seeding-steps
|   |   +-- sql
|   |   +-- types
|   |   +-- utils
|   |
|   +-- src
|       |
|       +-- controllers
|       +-- middleware
|       +-- routes
|       +-- types
|       +-- utils
|
+-- frontend
    |
    +-- public
    +-- src
        |
        +-- assets
        |   |
        |   +-- availableIcons
        |   +-- font
        |       |
        |       +-- jaldi
        |
        +-- components
        +-- pages
        +-- services
```

```
+-- stores
+-- styles
+-- utils
```

10 Installation

10.1 Manual

To start following this guide, please open XAMPP Control Panel and start the MySQL server. Then in a directory of your choosing, clone the repository from section 11 using

```
git clone https://github.com/JacopoFoschi2/ElaboratoISW.git
```

- OPTIONAL: if you want to use a more secure secret please create a .env file inside backend folder.
  Execute the command
  `node -e "console.log(require('crypto').randomBytes(64).toString('hex'))"`
  And copy the output. Then paste it in .env file following this structure

  ```
  JWT_SECRET=PASTE_HERE
  ```

  Following this step is strongly advised if the intention is to deploy the app in a prod setting. If you don't follow it the backend will work anyway without any issues but using an unsafe, hardcoded and public secret.
- open two terminal windows in the project root.
- on the first terminal change directory to backend with cd backend
- run the command npm run all and wait for it to end
- on the second terminal change directory to frontend with cd frontend
- run the command npm run all and wait for it to end
- open the shown http address (it should be https://localhost:4173)

11 Repository

11.1 Repo path

https://github.com/JacopoFoschi2/ElaboratoISW/tree/master