**Back-end:**

Configurações do Servidor:

const express = require('express')

require('express-async-errors');

const app = express()

const cors = require('cors')

const error = require('../middleware/error')

const user = require('../router/user')

const game = require('../router/game')

const platform = require('../router/platform')

app.use(cors())

app.use(express.json())

app.get('/', (req, res) => res.status(200).json({ message: 'ok' }))

app.use('/user', user)

app.use('/game', game)

app.use('/platform', platform)

app.use(error) //Próximos códigos acima dessa linha

module.exports = app

Inicialização do Servidor:

const app = require('./app')

app.listen(3001)

console.log('Api rodando na porta 3001')

Controladores “game”:

const gameService = require('../service/game');

async function insertGame(req, res) {

    await gameService.insertGame(req.body);

    res.status(200).json({ message: 'Jogo adicionado com sucesso!' });

}

async function updateGame(req, res) {

    await gameService.updateGame(req.params.id, req.body);

    res.status(200).json({ message: 'Jogo atualizado com sucesso!' });

}

async function deleteGame(req, res) {

    await gameService.deleteGame(req.params.id);

    res.status(200).json({ message: 'Jogo deletado com sucesso!' });

}

async function getAllGames(\_req, res) {

    const result = await gameService.getAllGames();

    res.status(200).json(result);

}

async function getPlatformByGames(req, res) {

    const result = await gameService.getPlatformByGames(req.params.id);

    res.status(200).json(result);

  }

module.exports = {

    insertGame, updateGame, deleteGame, getAllGames, getPlatformByGames

}

Controladores “plataform”:

const platformService = require('../service/platform');

async function insertPlatform(req, res) {

  await platformService.insertPlatform(req.body);

  res.status(200).json({ message: 'Plataforma adicionado com sucesso!' });

}

async function updatePlatform(req, res) {

  await platformService.updatePlatform(req.params.id, req.body);

  res.status(200).json({ message: 'Plataforma atualizado com sucesso!' });

}

async function deletePlatform(req, res) {

  await platformService.deletePlatform(req.params.id);

  res.status(200).json({ message: 'Plataforma deletado com sucesso!' });

}

async function getAllPlatform(\_req, res) {

  const result = await platformService.getAllPlatform();

  res.status(200).json(result);

}

async function addGameToPlatform(req, res) {

  const { idGame, idPlatform } = req.body;

  await platformService.addGameToPlatform(idGame, idPlatform);

  res.status(200).json({ message: 'Jogo adicinado com sucesso!'});

}

async function removeGameFromPlatform(req, res) {

  const { idGame, idPlatform } = req.body;

  await platformService.removeGameFromPlatform(idGame, idPlatform);

  res.status(200).json({ message: 'Jogo removido com sucesso!'});

}

async function getGamesByPlatform(req, res) {

  const result = await platformService.getGamesByPlatform(req.params.id);

  res.status(200).json(result);

}

module.exports = {

  insertPlatform,

  updatePlatform,

  deletePlatform,

  getAllPlatform,

  addGameToPlatform,

  removeGameFromPlatform,

  getGamesByPlatform,

}

Controladores “user”:

const userService = require('../service/user');

async function login(req, res) {

    const result = await userService.login(req.body);

    res.status(200).json(result);

}

async function getUser(req, res) {

    const result = await userService.getUser(req.body);

    res.status(200).json(result);

}

async function createUser(req, res) {

    const result = await userService.createUser(req.body);

    res.status(200).json(result);

}

async function updateUser(req, res) {

    await userService.updateUser(req.params.id, req.body);

    res.status(200).json({ message: 'Alteração feita com sucesso!' });

}

async function deleteUserById(req, res) {

    await userService.deleteUserById(req.params.id);

    res.status(200).json({ message: 'Usuário deletado com sucesso!' });

}

async function logicalUserDeletionById(req, res) {

    await userService.logicalUserDeletionById(req.params.id, req.body);

    res.status(200).json({ message: 'Usuário desativado com sucesso!' });

}

async function getGamesByUserId(req, res) {

    const result = await userService.getGamesByUserId(req.params.id);

    res.status(200).json(result);

}

async function addGameToUser(req, res) {

    const { userId, jogoId } = req.body;

    await userService.addGameToUser(userId, jogoId);

    res.status(200).json({ message: 'Jogo adicionado com sucesso!' });

}

async function removeGameFromUser(req, res) {

    const { userId, jogoId } = req.body;

    await userService.removeGameFromUser(userId, jogoId);

    res.status(200).json({ message: 'Jogo removido com sucesso!' });

}

module.exports = {

    login,

    getUser,

    createUser,

    updateUser,

    deleteUserById,

    logicalUserDeletionById,

    getGamesByUserId,

    addGameToUser,

    removeGameFromUser,

}

Configuração de conexão com DB:

require('dotenv').config();

const mysql = require('mysql2');

const connection = mysql.createConnection({

  host: process.env.MYSQL\_HOST,

  user: process.env.MYSQL\_USER,

  password: process.env.MYSQL\_PASSWORD,

  database: process.env.MYSQL\_DB\_NAME

});

connection.connect((err) => {

  if (err) {

    console.error('Erro ao conectar ao banco de dados:', err);

  } else {

    console.log('Conexão bem-sucedida ao banco de dados MySQL');

  }

});

module.exports = connection;

Criação do DB:

const connection = require('./connection');

connection.query('DROP DATABASE IF EXISTS dev\_games', (err) => {

  if (err) {

    console.error('Erro ao excluir o banco de dados dev\_games:', err);

  } else {

    console.log('Banco de dados "dev\_games" excluído com sucesso');

  }

});

connection.query('CREATE DATABASE IF NOT EXISTS dev\_games', (err) => {

  if (err) {

    console.error('Erro ao criar o banco de dados:', err);

  } else {

    console.log('Banco de dados criado com sucesso');

    // Usa o banco de dados

    connection.query('USE dev\_games', (err) => {

      if (err) {

        console.error('Erro ao usar o banco de dados dev\_games:', err);

      } else {

        console.log('Usando o banco de dados dev\_games');

        // Cria tabela User

        const createTableUser = `

          CREATE TABLE Users (

            id INT AUTO\_INCREMENT PRIMARY KEY,

            user VARCHAR(50) NOT NULL,

            firstName VARCHAR(50) NOT NULL,

            lastName VARCHAR(50) NOT NULL,

            email VARCHAR(255) NOT NULL,

            password VARCHAR(50) NOT NULL,

            userExit BOOLEAN DEFAULT 1

          );

        `;

        connection.query(createTableUser, (err) => {

          if (err) {

            console.error('Erro ao criar a tabela: User', err);

          } else {

            console.log('Tabela criada com sucesso User');

          }

        });

        // Cria tabela Plataforma

        const createTablePlataforma = `

          CREATE TABLE Plataforma (

            id INT AUTO\_INCREMENT PRIMARY KEY,

            nome VARCHAR(255) NOT NULL,

            INDEX (nome)  -- Add an index on the 'nome' column

          );

        `

        connection.query(createTablePlataforma, (err) => {

          if (err) {

            console.error('Erro ao criar a tabela Plataforma:', err);

          } else {

            console.log('Tabela criada com sucesso Plataforma');

          }

        });

        // Cria tabela Jogo

        const createTableJogos = `

          CREATE TABLE Jogo (

            id INT AUTO\_INCREMENT PRIMARY KEY,

            nome VARCHAR(255) NOT NULL,

            cat VARCHAR(255),

            plataforma\_que VARCHAR(255),

            nota DECIMAL(3,1),

            status ENUM('jogando', 'jogado', 'zerado', 'outros'),

            recomendacao ENUM('Sim', 'Não'),

            INDEX (nome)  -- Add an index on the 'nome' column

          );

        `

        connection.query(createTableJogos, (err) => {

          if (err) {

            console.error('Erro ao criar a tabela Jogos:', err);

          } else {

            console.log('Tabela criada com sucesso Jogos');

          }

        });

        // Cria tabela Jogo\_Plataforma

        const createTableJogoPlataforma = `

          CREATE TABLE Jogo\_Plataforma (

            id INT AUTO\_INCREMENT PRIMARY KEY,

            id\_jogo INT NOT NULL,

            id\_plataforma INT NOT NULL,

            FOREIGN KEY (id\_jogo) REFERENCES Jogo(id),

            FOREIGN KEY (id\_plataforma) REFERENCES Plataforma(id),

            UNIQUE KEY unique\_jogo\_plataforma (id\_jogo, id\_plataforma)

          );

        `

        connection.query(createTableJogoPlataforma, (err) => {

          if (err) {

            console.error('Erro ao criar a tabela Jogo\_Plataforma:', err);

          } else {

            console.log('Tabela criada com sucesso Jogo\_Plataforma');

          }

        });

        // Cria tabela User\_Jogo

        const createTableUserJogo = `

          CREATE TABLE User\_Jogo (

            id INT AUTO\_INCREMENT PRIMARY KEY,

            userId INT,

            jogoId INT,

            FOREIGN KEY (userId) REFERENCES Users(id),

            FOREIGN KEY (jogoId) REFERENCES Jogo(id),

            UNIQUE KEY unique\_user\_jogo (userId, jogoId)

          );

        `

        connection.query(createTableUserJogo, (err) => {

          if (err) {

            console.error('Erro ao criar a tabela User\_Jogo:', err);

          } else {

            console.log('Tabela criada com sucesso User\_Jogo');

          }

        });

        const insertUsers = `

          INSERT INTO Users (user, firstName, lastName, email, password, userExit) VALUES

            ('user1', 'John', 'Doe', 'john.doe@email.com', '5f4dcc3b5aa765d61d8327deb882cf99', true),

            ('user2', 'Jane', 'Smith', 'jane.smith@email.com', '5f4dcc3b5aa765d61d8327deb882cf99', true),

            ('user3', 'Bob', 'Johnson', 'bob.johnson@email.com', '5f4dcc3b5aa765d61d8327deb882cf99', true),

            ('user4', 'Alice', 'Jones', 'alice.jones@email.com', '5f4dcc3b5aa765d61d8327deb882cf99', true),

            ('user5', 'Charlie', 'Brown', 'charlie.brown@email.com', '5f4dcc3b5aa765d61d8327deb882cf99', true),

            ('user6', 'Eva', 'Miller', 'eva.miller@email.com', '5f4dcc3b5aa765d61d8327deb882cf99', true),

            ('user7', 'David', 'Williams', 'david.williams@email.com', '5f4dcc3b5aa765d61d8327deb882cf99', true),

            ('user8', 'Sophia', 'Davis', 'sophia.davis@email.com', '5f4dcc3b5aa765d61d8327deb882cf99', true),

            ('user9', 'Michael', 'Moore', 'michael.moore@email.com', '5f4dcc3b5aa765d61d8327deb882cf99', true),

            ('user10', 'Olivia', 'Taylor', 'olivia.taylor@email.com', '5f4dcc3b5aa765d61d8327deb882cf99', true);

          `

        // Inserindo Usuarios

        connection.query(insertUsers, (err) => {

          if (err) {

            console.error('Erro ao inserir dados na tabela Users:', err);

          } else {

            console.log('Dados inseridos na tabela Users com sucesso');

          }

        });

        const insertPlataforma = `

          INSERT INTO Plataforma (nome) VALUES

            ('Steam'),

            ('Xbox Store'),

            ('Uplay'),

            ('GOG'),

            ('Epic Games Store'),

            ('PlayStation Store'),

            ('G2A'),

            ('Green Man Gaming'),

            ('Humble Store'),

            ('Origin')

          `;

        // Inserindo plataformas

        connection.query(insertPlataforma, (err) => {

          if (err) {

            console.error('Erro ao inserir dados na tabela Plataforma:', err);

          } else {

            console.log('Dados inseridos na tabela Plataforma com sucesso');

          }

        });

        const insertJogos = `

        INSERT INTO Jogo (nome, cat, plataforma\_que, nota, status, recomendacao) VALUES

        ('The Witcher 3', 'RPG', 'Epic Games Store', 9.5, 'zerado', 'Sim'),

        ('Red Dead Redemption 2', 'Ação e Aventura', 'Steam', 9.8, 'zerado', 'Sim'),

        ('Animal Crossing: New Horizons', 'Simulação', 'Uplay', 9.0, 'jogado', 'Sim'),

        ('Cyberpunk 2077', 'RPG', 'Steam', 8.0, 'outros', 'Não'),

        ('Among Us', 'Multijogador', 'Steam', 8.5, 'jogado', 'Sim'),

        ('The Legend of Zelda: Breath of the Wild', 'Ação e Aventura', 'PlayStation Store', 9.7, 'zerado', 'Sim'),

        ('GTA V', 'Ação e Aventura', 'PC', 9.3, 'zerado', 'Sim'),

        ('FIFA 23', 'Esportes', 'PlayStation Store', 8.2, 'jogado', 'Sim'),

        ('Minecraft', 'Aventura', 'Epic Games Store', 9.0, 'zerado', 'Sim'),

        ('Super Mario Odyssey', 'Ação e Aventura', 'Uplay', 9.5, 'zerado', 'Sim');

          `

        // Inserindo Jogos

        connection.query(insertJogos, (err) => {

          if (err) {

            console.error('Erro ao inserir dados na tabela Jogos:', err);

          } else {

            console.log('Dados inseridos na tabela Jogos com sucesso');

          }

        });

        const insertJogoPlataforma = `

          INSERT INTO Jogo\_Plataforma (id\_jogo, id\_plataforma) VALUES

            (1, 1),

            (1, 4),

            (1, 5),

            (2, 1),

            (2, 2),

            (3, 3),

            (4, 4),

            (4, 1),

            (5, 7),

            (6, 3),

            (6, 8),

            (7, 4),

            (7, 1),

            (7, 2),

            (8, 5),

            (8, 6),

            (8, 9),

            (8, 10),

            (9, 4),

            (9, 1),

            (9, 2),

            (9, 3),

            (10, 3),

            (10, 7);

          `

        // Inserindo Relacinamento Jogo\_Plataforma

        connection.query(insertJogoPlataforma, (err) => {

          if (err) {

            console.error('Erro ao inserir dados na tabela Jogo\_Plataforma:', err);

          } else {

            console.log('Dados inseridos na tabela Jogo\_Plataforma com sucesso');

          }

        });

        // Inserindo Relacinamento User\_Jogo

        const insertUserJogo = `

          INSERT INTO User\_Jogo (userId, jogoId) VALUES

            (1, 1),

            (1, 2),

            (2, 3),

            (2, 4),

            (3, 5),

            (3, 6),

            (4, 7),

            (4, 8),

            (5, 9),

            (5, 10),

            (6, 1),

            (6, 2),

            (7, 3),

            (7, 4),

            (8, 5),

            (8, 6),

            (9, 7),

            (9, 8),

            (10, 9),

            (10, 10);

        `

        connection.query(insertUserJogo, (err) => {

          if (err) {

            console.error('Erro ao inserir dados na tabela Jogo\_Plataforma:', err);

          } else {

            console.log('Dados inseridos na tabela Jogo\_Plataforma com sucesso');

          }

        });

        // Fecha a conexão

        connection.end();

      }

    });

  }

});

CRUD “game”:

const connection = require('./connection');

async function insertGame(data) {

  return new Promise(function (resolve, reject) {

    const { nome, cat, plataforma\_que, nota, status, recomendacao} = data;

    const query = 'INSERT INTO Jogo (nome, cat, plataforma\_que, nota, status, recomendacao) VALUES (?, ?, ?, ?, ?, ?)'

    connection.query(query, [nome, cat, plataforma\_que, nota, status, recomendacao], (err, results) => {

      if (err) {

        reject(err);

      } else {

        resolve(results);

      }

    });

  });

}

async function updateGame(gameId, game) {

  return new Promise((resolve, reject) => {

    const { nome, cat, plataforma\_que, nota, status, recomendacao } = game;

    const query = 'UPDATE Jogo SET nome = ?, cat = ?, plataforma\_que = ?, nota = ?, status = ?, recomendacao = ? WHERE id = ?';

    connection.query(query, [nome, cat, plataforma\_que, nota, status, recomendacao, gameId], (err, results) => {

      if (err) {

        reject(err);

      } else {

        if (results.affectedRows > 0) {

          resolve(results);

        } else {

          reject(new Error('Nenhum jogo encontrado para atualização'));

        }

      }

    });

  });

}

async function deleteGame(gameId) {

  return new Promise((resolve, reject) => {

    const query = 'DELETE FROM Jogo WHERE id = ?';

    connection.query(query, [gameId], (err, results) => {

      if (err) {

        reject(err);

      } else {

        if (results.affectedRows > 0) {

          resolve('Jogo deletado com sucesso');

        } else {

          console.log(`Delet erro capiturado: ${err}`);

          reject(new Error('Nenhum jogo encontrado para exclusão'));

        }

      }

    });

  });

}

async function getAllGames() {

  return new Promise((resolve, reject) => {

    const query = `SELECT

    Jogo.id,

    Jogo.nome,

    Jogo.cat,

    Jogo.nota,

    Jogo.status,

    Jogo.recomendacao,

    GROUP\_CONCAT(Plataforma.nome SEPARATOR ', ') AS plataforma\_disp

FROM

    Jogo

LEFT JOIN

    Jogo\_Plataforma ON Jogo.id = Jogo\_Plataforma.id\_jogo

LEFT JOIN

    Plataforma ON Jogo\_Plataforma.id\_plataforma = Plataforma.id

GROUP BY

    Jogo.id`;

    connection.query(query, (err, results) => {

      if (err) {

        reject(err);

      } else {

        resolve(results);

      }

    });

  });

}

async function getPlatformByGames(idGame) {

  return new Promise((resolve, reject) => {

    const query =

      `SELECT P.nome AS nome\_plataforma

      FROM Jogo\_Plataforma JP

      INNER JOIN Plataforma P ON JP.id\_plataforma = P.id

      WHERE JP.id\_jogo = ?`

      ;

    connection.query(query, [idGame], (err, results) => {

      if (err) {

        reject(err);

      } else {

        resolve(results);

      }

    });

  });

}

module.exports = {

  insertGame, updateGame, deleteGame, getAllGames, getPlatformByGames

}

CRUD “plataform”:

const connection = require('./connection');

async function insertPlatform(data) {

  return new Promise(function (resolve, reject) {

    const { nome } = data;

    const query = 'INSERT INTO Plataforma (nome) VALUES (?)'

    connection.query(query, [nome], (err, results) => {

      if (err) {

        reject(err);

      } else {

        resolve(results);

      }

    });

  });

}

async function updatePlatform(id, data) {

  return new Promise((resolve, reject) => {

    const { nome } = data;

    const query = `

      UPDATE Plataforma SET nome = ? WHERE id = ?

    `;

    connection.query(query, [nome, id], (err, results) => {

      if (err) {

        reject(err);

      } else {

        if (results.affectedRows > 0) {

          resolve(results);

        } else {

          reject(new Error('Nenhuma Plataforma encontrado para atualização'));

        }

      }

    });

  });

}

async function deletePlatform(gameId) {

  return new Promise((resolve, reject) => {

    const query = 'DELETE FROM Plataforma WHERE id = ?';

    connection.query(query, [gameId], (err, results) => {

      if (err) {

        reject(err);

      } else {

        if (results.affectedRows > 0) {

          resolve('Plataforma deletado com sucesso');

        } else {

          reject(new Error('Nenhuma Plataforma encontrado para exclusão'));

        }

      }

    });

  });

}

async function getAllPlatform() {

  return new Promise((resolve, reject) => {

    const query = 'SELECT \* FROM Plataforma';

    connection.query(query, (err, results) => {

      if (err) {

        reject(err);

      } else {

        resolve(results);

      }

    });

  });

}

async function addGameToPlatform(idGame, idPlatform) {

  return new Promise((resolve, reject) => {

    const query = 'INSERT INTO Jogo\_Plataforma (id\_jogo, id\_plataforma) VALUES (?, ?)';

    connection.query(query, [idGame, idPlatform], (err, results) => {

      if (err) {

        reject(err);

      } else {

        resolve(results);

      }

    });

  });

}

async function removeGameFromPlatform(idGame, idPlatform) {

  return new Promise((resolve, reject) => {

    const query = 'DELETE FROM Jogo\_Plataforma WHERE id\_plataforma = ? AND id\_jogo = ?';

    connection.query(query, [idPlatform, idGame], (err, \_results) => {

      if (err) {

        reject(err);

      } else {

        resolve();

      }

    });

  });

}

async function getGamesByPlatform(idPlatform) {

  return new Promise((resolve, reject) => {

    const query = `

      SELECT Jogo.id, Jogo.nome

      FROM Jogo

      INNER JOIN Jogo\_Plataforma ON Jogo.id = Jogo\_Plataforma.id\_jogo

      WHERE Jogo\_Plataforma.id\_plataforma = ?;

    `;

    connection.query(query, [idPlatform], (err, results) => {

      if (err) {

        reject(err);

      } else {

        resolve(results);

      }

    });

  });

}

module.exports = {

  insertPlatform,

  updatePlatform,

  deletePlatform,

  getAllPlatform,

  addGameToPlatform,

  removeGameFromPlatform,

  getGamesByPlatform,

}

CRUD “user”:

const connection = require('./connection');

async function getUser(user) {

  return new Promise((resolve, reject) => {

    const query = 'SELECT \* FROM Users WHERE user = ? AND  userExit = true';

    connection.query(query, [user], (err, results) => {

      if (err) {

        reject(err);

      } else {

        if (results.length > 0) {

          resolve(results[0]);

        } else {

          resolve(null);

        }

      }

    });

  });

}

async function insertUser(data) {

  return new Promise((resolve, reject) => {

    const { user, firstName, lastName, email, password } = data;

    const query = 'INSERT INTO Users (user, firstName, lastName, email, password) VALUES (?, ?, ?, ?, ?)';

    connection.query(query, [user, firstName, lastName, email, password], (err, results) => {

      if (err) {

        reject(err);

      } else {

        resolve(results.insertId);

      }

    });

  });

}

async function patchUser(id, updateUser) {

  return new Promise((resolve, reject) => {

    const {

      user, firstName, lastName, email, password

    } = updateUser

    const query = 'UPDATE Users SET user = ?, firstName = ?, lastName = ?, email = ?, password = ? WHERE id = ?';

    connection.query(query, [user, firstName, lastName, email, password, id], (err, results) => {

      if (err) {

        reject(err);

      } else {

        if (results.affectedRows > 0) {

          resolve(results[0]);

        } else {

          reject(new Error('Nenhum usuário encontrado para atualização'));

        }

      }

    });

  });

}

async function deleteUserById(userId) {

  return new Promise((resolve, reject) => {

    const query = 'DELETE FROM Users WHERE id = ?';

    connection.query(query, [userId], (err, results) => {

      if (err) {

        reject(err);

      } else {

        if (results.affectedRows > 0) {

          resolve();

        } else {

          reject(new Error('Nenhum usuário encontrado para deleção'));

        }

      }

    });

  });

}

async function logicalUserDeletionById(id, delit) {

  return new Promise((resolve, reject) => {

    const query = 'UPDATE Users SET userExit = ? WHERE id = ?';

    connection.query(query, [delit, id], (err, results) => {

      if (err) {

        reject(err);

      } else {

        if (results.affectedRows > 0) {

          resolve(results[0]);

        } else {

          reject(new Error('Nenhum usuário encontrado para atualização'));

        }

      }

    });

  });

}

function getGamesByUserId(userId) {

  return new Promise((resolve, reject) => {

    const query = `

      SELECT Jogo.\*

      FROM Jogo

      INNER JOIN User\_Jogo ON Jogo.id = User\_Jogo.jogoId

      WHERE User\_Jogo.userId = ?;

    `;

    connection.query(query, [userId], (err, results) => {

      if (err) {

        reject(err);

      } else {

        resolve(results);

      }

    });

  });

}

function addGameToUser(userId, jogoId) {

  return new Promise((resolve, reject) => {

    const query = 'INSERT INTO User\_Jogo (userId, jogoId) VALUES (?, ?)';

    connection.query(query, [userId, jogoId], (err, results) => {

      if (err) {

        reject(err);

      } else {

        resolve(results);

      }

    });

  });

}

function removeGameFromUser(userId, jogoId) {

  return new Promise((resolve, reject) => {

    const query = 'DELETE FROM User\_Jogo WHERE userId = ? AND jogoId = ?';

    connection.query(query, [userId, jogoId], (err, results) => {

      if (err) {

        reject(err);

      } else {

        resolve();

      }

    });

  });

}

module.exports = {

  getUser,

  insertUser,

  patchUser,

  deleteUserById,

  logicalUserDeletionById,

  getGamesByUserId,

  addGameToUser,

  removeGameFromUser,

};

Rota “game”:

const router = require('express').Router()

const gameController = require('../controller/game')

router.post('/insert', gameController.insertGame)

router.patch('/update/:id', gameController.updateGame)

router.delete('/delete/:id', gameController.deleteGame)

router.get('/', gameController.getAllGames)

router.get('/:id/platform', gameController.getPlatformByGames)

module.exports = router

Rota “platform”:

const router = require('express').Router()

const platformController = require('../controller/platform')

router.post('/insert', platformController.insertPlatform)

router.patch('/update/:id', platformController.updatePlatform)

router.delete('/delete/:id', platformController.deletePlatform)

router.get('/', platformController.getAllPlatform)

router.get('/:id/games', platformController.getGamesByPlatform)

router.post('/game', platformController.addGameToPlatform)

router.delete('/delete-game', platformController.removeGameFromPlatform)

module.exports = router

Rota “user”:

const router = require('express').Router()

const userController = require('../controller/user')

router.post('/create', userController.createUser)

router.post('/login', userController.login)

router.post('/', userController.getUser)

router.patch('/update/:id', userController.updateUser)

router.delete('/delete/:id', userController.deleteUserById)

router.patch('/disabled/:id', userController.logicalUserDeletionById)

router.get('/games/:id', userController.getGamesByUserId)

router.post('/game', userController.addGameToUser)

router.post('/delete-game', userController.removeGameFromUser)

module.exports = router

Service “game”:

const gameModel = require('../models/game');

async function insertGame(data) {

  try {

    const result = await gameModel.insertGame(data);

    return result;

  } catch (error) {

    throw new Error('400|Erro ao cadastrar: Verifique se todos os campos estão preenchidos corretamente.')

  }

}

async function updateGame(gameId, game) {

  try {

    const result = await gameModel.updateGame(gameId, game);

  return result;

  } catch (error) {

    throw new Error('400|Erro ao atualizar: Verifique se todos os campos estão preenchidos corretamente.')

  }

}

async function deleteGame(gameId) {

  try {

    const result = await gameModel.deleteGame(gameId);

    return result;

  } catch (error) {

    throw new Error('409|Jogo vinculado a uma Jogador e/ou Plataforma')

  }

}

async function getAllGames() {

  const result = await gameModel.getAllGames();

  return result;

}

async function getPlatformByGames(idGame) {

  const result = await gameModel.getPlatformByGames(idGame);

  return result;

}

module.exports = {

  insertGame,

  updateGame,

  deleteGame,

  getAllGames,

  getPlatformByGames

}

Service “platform”:

const platformModels = require('../models/platform');

async function insertPlatform(data) {

  const result = platformModels.insertPlatform(data);

  return result;

}

async function updatePlatform(id, data) {

  await platformModels.updatePlatform(id, data);

}

async function deletePlatform(id){

  try {

    await platformModels.deletePlatform(id);

  } catch (error) {

    throw new Error('409|Plataforma vinculado a uma jogo.')

  }

}

async function getAllPlatform() {

  return await platformModels.getAllPlatform();

}

async function addGameToPlatform(idGame, idPlatform) {

  try {

      if (!idGame || !idPlatform) throw new Error('400|idPlatform e idGame são obrigatórios.');

      await platformModels.addGameToPlatform(idGame, idPlatform);

  } catch (error) {

      throw new Error('409|O jogo já existe nessa Plataforma e não pode ser duplicado.');

  }

}

async function removeGameFromPlatform(idGame, idPlatform) {

  await platformModels.removeGameFromPlatform(idGame, idPlatform);

}

async function getGamesByPlatform(idPlatform) {

  return await platformModels.getGamesByPlatform(idPlatform);

}

module.exports = {

  insertPlatform,

  updatePlatform,

  deletePlatform,

  getAllPlatform,

  addGameToPlatform,

  removeGameFromPlatform,

  getGamesByPlatform,

}

Service “user”:

const userModel = require('../models/user')

const generateJWT = require('../util/generatejwt')

const md5 = require('md5')

async function login({ user, password }) {

  const verify = await userModel.getUser(user)

  if (!verify) {

    throw new Error('404|Esse usuário não existe')

  }

  if (md5(password).localeCompare(verify.password)) {

    throw new Error('401|Senha incorreta')

  }

  const newUser = {

    id: verify.id,

    firstName: verify.firstName,

    lastName: verify.lastName,

    password: verify.password,

    email: verify.email,

    user: verify.user,

    token: generateJWT(verify)

  }

  return newUser

}

async function getUser({user}) {

  try {

  const result = await userModel.getUser(user)

  return result

  } catch (error) {

    throw new Error('400|Usuário não encontrado')

  }

}

async function createUser(data) {

  const verify = await userModel.getUser(data.user)

  if (verify) {

    throw new Error('409|Usuário já existente')

  }

  try {

    await userModel.insertUser({ ...data, password: md5(data.password) })

    return { message: 'Cadastro efetuado com sucesso' }

  } catch (error) {

    throw new Error(`500|${error.message}`)

  }

}

async function updateUser(id, updateUser) {

  const result = await userModel.patchUser(id, { ...updateUser, password: md5(updateUser.password) });

  return result;

}

async function deleteUserById(id) {

  await userModel.deleteUserById(id);

}

async function logicalUserDeletionById(id, delit) {

  await userModel.logicalUserDeletionById(id, delit);

}

async function getGamesByUserId(userId) {

  const result = await userModel.getGamesByUserId(userId);

  return result;

}

async function addGameToUser(userId, jogoId) {

  try {

    if (!userId || !jogoId) throw new Error('400|userId e jogoId são obrigatórios.');

    await userModel.addGameToUser(userId, jogoId);

  } catch (error) {

    throw new Error('409|Usuário já possui este jogo.');

  }

}

async function removeGameFromUser(userId, jogoId) {

  await userModel.removeGameFromUser(userId, jogoId);

}

module.exports = {

  login,

  getUser,

  createUser,

  updateUser,

  deleteUserById,

  logicalUserDeletionById,

  getGamesByUserId,

  addGameToUser,

  removeGameFromUser,

}

Middleware token:

const jwt = require('jsonwebtoken');

const jwtKey = 'minha\_senha';

const auth = (req, \_res, next) => {

  const token = req.headers.authorization;

  if (!token) throw new Error('404|Token não encontrado!');

  try {

    jwt.verify(token, jwtKey);

    next();

  } catch (error) {

    throw new Error('401|Token inválido!');

  }

};

module.exports = auth;

Middleware err:

const erros = (err, \_req, res, \_next) => {

console.log('error', err)

  const [code, message] = err.message.split('|');

  if (code && message) {

    return res.status(Number(code)).json({ message });

  }

  console.log(err);

  return res.status(500)

    .json({ message: 'Internal server error' });

};

module.exports = erros;

**Front-end:**

Biblioteca:

'use client'

import { CardBiblioteca } from "@/components/cardBiblioteca";

import NavBarHome from "@/components/navBarHome";

import { getGamesAxios } from "@/util/axios";

import { useEffect, useState } from "react";

export default function Biblioteca() {

  const [games, setGames] = useState([]);

  const [user, setUser] = useState({});

  useEffect(() => {

    const user = JSON.parse(localStorage.getItem('user'));

    setUser(user)

  }, [])

  useEffect(() => {

    const fetchData = async () => {

      try {

        const result = await getGamesAxios(user.id);

        setGames(result);

      } catch (error) {

        console.error('Error fetching data:', error);

      }

    };

    fetchData();

  }, [user.id, games]);

  return (

    <main>

      <NavBarHome />

      <div className="home-section">

        {games && games.map(({ id, cat, nome, nota, plataforma\_disp, recomendacao, status }) => (

          <CardBiblioteca

            key={id}

            userId={user.id}

            id={id}

            cat={cat}

            nome={nome}

            nota={nota}

            plataforma\_disp={plataforma\_disp}

            recomendacao={recomendacao}

            status={status}

          />

        ))}

      </div>

    </main>

  );

}

Cadastro:

'use client'

import { ListarJogos } from "@/components/listaJogos"

import { ListaPlataforma } from "@/components/listaPlataforma"

import NavBarHome from "@/components/navBarHome"

import { useState } from "react"

export default function Cadastro() {

  const [render, setRender] = useState(true)

  return (

    <main>

      <NavBarHome />

      <div className="cadastro-main">

        <div className="cadastro-div">

          <span className="cadastro-span" onClick={() => setRender(true)}>Jogo</span>

          <span className="cadastro-span" onClick={() => setRender(false)}>Plataforma</span>

        </div>

        {

          render ? <ListarJogos /> : <ListaPlataforma />

        }

      </div>

    </main>

  )

}

Comunidade:

'use client'

import NavBar from '@/components/navbar';

import { useEffect, useRef, useState } from 'react';

export default function Comunidade() {

  const [messages, setMessages] = useState([]);

  const [newMessage, setNewMessage] = useState('');

  const chatMessagesRef = useRef(null);

  const handleSendMessage = () => {

    if (newMessage.trim() !== '') {

      setMessages([...messages, { text: newMessage, timestamp: new Date() }]);

      setNewMessage('');

    }

  };

  useEffect(() => {

    if (chatMessagesRef.current) {

      chatMessagesRef.current.scrollTop = chatMessagesRef.current.scrollHeight;

    }

  }, [messages]);

  return (

    <div>

      <NavBar />

      <div>

        <div className="chat-messages" ref={chatMessagesRef}>

          {messages.map((message, index) => (

            <div key={index}>

              {message.text} - {message.timestamp.toLocaleTimeString()}

            </div>

          ))}

        </div>

      </div>

      <div className='div-chat'>

        <input

          type="text"

          value={newMessage}

          className="input-chat-message"

          onChange={(e) => setNewMessage(e.target.value)}

          placeholder="Digite sua messagen..."

        />

        <button

          className="input-chat-btn"

          onClick={handleSendMessage}

        >

          Enviar

        </button>

      </div>

    </div>

  );

};

Criar Usuário:

'use client'

import NavBar from "@/components/navbar";

import { createUserAxios } from "@/util/axios";

import { useState } from "react";

export default function CreateUser() {

  const [firstName, setFirstName] = useState('')

  const [lastName, setLastNamee] = useState('')

  const [password, setPassword] = useState('')

  const [password1, setPassword1] = useState('')

  const [email, setEmail] = useState('')

  const [user, setUser] = useState('')

  const createUser = async () => {

    try {

      const newUser = { firstName, lastName, email, password, user }

      if (password <= 5) return alert('Senha deve ter 6 caracteres')

      if (password !== password1) return alert('Senhas devem ser iguais')

      const result = await createUserAxios(newUser)

      if (result) alert(result)

    } catch (error) {

      alert(error.response.data.message)

    }

  }

  return (

    <section className="area-create">

      <NavBar />

      <div className="create">

          <h3 className="titulo" >Cadastrar</h3>

      <form className="creatUser-form">

      <input

          type="text"

          value={email}

          name="email"

          className="create-input"

          placeholder="Endereço de e-mail"

          onChange={({ target }) => setEmail(target.value)}

          required

        />

        <input

          type="text"

          value={firstName}

          name="firstName"

          className="create-input"

          placeholder="Nome"

          onChange={({ target }) => setFirstName(target.value)}

          required

        />

        <input

          type="text"

          value={lastName}

          name="lastName"

          className="create-input"

          placeholder="Sobrenome"

          onChange={({ target }) => setLastNamee(target.value)}

          required

        />

        <input

          type="text"

          value={user}

          name="user"

          className="create-input"

          placeholder="Usuário"

          onChange={({ target }) => setUser(target.value)}

          required

        />

        <input

          type="password"

          value={password}

          name="password"

          className="create-input"

          placeholder="Senha"

          onChange={({ target }) => setPassword(target.value)}

          required

        />

        <input

          type="password"

          value={password1}

          name="password1"

          className="create-input"

          placeholder="Confirma senha"

          onChange={({ target }) => setPassword1(target.value)}

          required

        />

        <input

          type="button"

          value="Cadastrar"

          onClick={createUser}

          className="create-button"

        />

      </form>

      </div>

    </section>

  )

}

Esqueceu Senha:

'use client'

import NavBar from "@/components/navbar";

import { getUserAxios } from "@/util/axios";

import { sendEmail } from "@/util/sendEmail";

import Link from "next/link";

import { useState } from "react";

export default function EsqueceuSenha() {

  const [user, setName] = useState("")

  const handleSubmit = async (e) => {

    e.preventDefault();

    try {

      const result = await getUserAxios({ user })

      if(!result) return alert('Usuário não encontrado!')

      const response = await sendEmail(result.email, result.user)

      if(response) alert('Email enviado com sucesso! Verifique o spam.')

    } catch (error) {

      if(error.response)

        alert(error.response.data.message)

    }

  }

  return (

    <section className="area-recuperar-senha">

      <NavBar/>

      <div className="recuperar">

        <div className="recuperar-img">

        <img src="/img/logo.png" alt="logo" />

        </div>

      <p id="p2">Enviaremos um e-mail de recuperação de senha para o e-mail cadastrado.</p>

      <form className="recuperar-senha">

        <input

          type="text"

          name="nome"

          className="recuperar-input"

          value={user}

          placeholder="nome de usuario"

          onChange={({ target }) => setName(target.value)}

        />

        <span className="recuperar-span">Digite seu usuário</span>

        <input type="submit" className="recuperar-botao" value="Enviar" onClick={handleSubmit} />

      </form>

      <p id="p3">Já tem uma conta?<Link href="/login">Fazer login</Link></p>

      </div>

    </section>

  )

}

Esqueceu senha [user]:

'use client'

import { getUserAxios, updateUserAxios } from "@/util/axios"

import { useEffect, useState } from "react"

export default function Recuperar({ params }) {

    const [user, setUser] = useState()

    const [password, setPassword] = useState('')

    const [password1, setPassword1] = useState('')

    useEffect(() => {

        const fetchData = async () => {

            const user = params.user

            const result = await getUserAxios({ user })

            setUser(result)

        }

        fetchData()

    }, [])

    const atualizar = async () => {

        try {

            if (password !== password1) return alert('Senhas devem ser iguais')

            const result = await updateUserAxios(user.id, { ...user, password })

            if (result) return alert(result)

        } catch (error) {

            alert(error.response.data.message)

        }

    }

    return (

        <form>

            <input

                type="password"

                name="password"

                value={password}

                onChange={({ target }) => setPassword(target.value)}

                placeholder="Insira uma nova senha"

            />

            <input

                type="password"

                name="password1"

                value={password1}

                onChange={({ target }) => setPassword1(target.value)}

                placeholder="Insira uma nova senha"

            />

            <input

                type="button"

                name="Enviar"

                value="Atualizar"

                onClick={atualizar}

            />

        </form>

    )

}

Home:

'use client'

import { Card } from "@/components/card";

import NavBarHome from "@/components/navBarHome";

import { gamesAxios } from "@/util/axios";

import { useEffect, useState } from "react";

export default function Home() {

  const [games, setGames] = useState([]);

  const [user, setUser] = useState({});

  useEffect(() => {

    const fetchData = async () => {

      try {

        const result = await gamesAxios();

        setGames(result);

        setUser(JSON.parse(localStorage.getItem('user')));

      } catch (error) {

        console.error('Error fetching data:', error);

      }

    };

    fetchData();

  }, []);

  return (

    <section>

      <NavBarHome />

      <div className="home-section">

        {games.map(({ id, cat, nome, nota, plataforma\_disp, recomendacao, status }) => (

          <Card

            key={id}

            userId={user.id}

            id={id}

            cat={cat}

            nome={nome}

            nota={nota}

            plataforma\_disp={plataforma\_disp}

            recomendacao={recomendacao}

            status={status}

          />

        ))}

      </div>

    </section>

  );

}

Login:

'use client'

import { loginAxios } from "@/util/axios"

import { useRouter } from "next/navigation"

import { useState } from "react"

import { createHash } from 'crypto'

import { setCookie } from 'cookies-next'

import Link from "next/link";

import NavBar from "@/components/navbar"

export default function Login() {

  const [user, setName] = useState("")

  const [password, setPassword] = useState("")

  const { push } = useRouter()

  const login = async (e) => {

    e.preventDefault()

    try {

      if (user.length < 3) {

        alert("Nome deve ter no mínimo 3 caracteres")

        return

      }

      if (password.length < 6) {

        alert("Digite uma senha com no minimo 6 caracteres")

        return

      }

      const result = await loginAxios({ user, password })

      if (result.user === user && result.password === createHash('md5').update(password).digest('hex')) {

        localStorage.setItem('user', JSON.stringify(result))

        setCookie('user', JSON.stringify(result))

        push("/home")

      }

    } catch (error) {

      alert(error.response.data.message)

    }

  }

  return (

    <section className="area-login">

      <NavBar/>

    <div className="login">

    <div>

      <img src="/img/logo.png" />

    </div>

    <form className="login-form">

      <input

        type="text"

        name="nome"

        value={user}

        className="login-input"

        placeholder="nome de usuario"

        onChange={({ target }) => setName(target.value)}

      />

      <span className="login-span">

        Nome deve ter no mínimo 5 caracteres

      </span>

      <input

        type="password"

        name="senha"

        value={password}

        className="login-input"

        placeholder="sua senha"

        onChange={({ target }) => setPassword(target.value)}

      />

      <span className="login-span">

        Digite uma senha com no minimo 6 caracteres

      </span>

      <input

        type="submit"

        value="Entrar"

        className="login-input-submit"

        onClick={login}

      />

    </form>

    <Link id="a1" href="esqueceuSenha">Esqueceu sua senha?</Link>

    <p id="p1" >Ainda não tem uma conta?<Link href="/createUser">Criar conta</Link></p>

    </div>

    </section>

  )

}

Notícias:

import NavBar from "@/components/navbar";

export default function Noticias() {

  return (

    <section>

      <NavBar />

      <div className="noticias">

        <h1 className="noticias-h1">Notícias</h1>

        <br />

        <h4>Fique por dentro das noticias dos jogos do momento!</h4>

        <br />

        <h2>Red Dead Redemption 2 </h2>

        <br />

        <h3>Red Dead 2 bate recorde de jogadores antes de trailer de GTA 6</h3>

        <br />

        <p>Red Dead Redemption 2 bateu recorde de jogadores simultâneos na Steam neste último final de semana! Curiosamente, a marca foi alcançada poucos dias antes do aguardado trailer do novo GTA 6, previsto para dezembro.</p>

        <br />

        <h3>Red Dead Redemption 2 faz aniversário e entra em promoção! Veja preços </h3>

        <br />

        <p>Na quarta-feira, dia 26 de outubro de 2023, faz 5 anos que Red Dead Redemption 2 foi lançado originalmente no PlayStation 4 e Xbox One. Para comemorar essa data tão especial, o game está com uma ótima promoção em algumas plataformas, chegando a um desconto de mais de 60% dependendo de onde você o adquirir.</p>

        <br />

        <h2>The Witcher 3</h2>

        <br />

        <h3>Nova atualização de The Witcher 3 será lançada em breve - Veja as mudanças</h3>

        <br />

        <p>Por meio do seu site oficial a CD Projekt RED revelou que uma nova atualização (Patch 4.03) será lançado muito em breve para The Witcher 3 e contará com grandes melhorias.</p>

        <p>A atualização se aplica no PC, PlayStation e Xbox e tem como objetivo corrigir problemas de jogabilidade, missões e mais. Além disso, o estúdio comenta sobre a atualização no Nintendo Switch.</p>

        <h4>Detalhes do Patch:</h4>

        <br />

        <strong>Especifico para PC</strong>

        <p>Correção de um problema com as sombras da vegetação com traçado de raios “estourando”, dependendo da proximidade e do ângulo do Geralt.

          Correção de um problema em que a desativação e a ativação repetida de reflexos e sombras com traçado de raios poderia causar a ativação de sombras pretas na borda dos objetos.

          Resolução do problema em que a versão DirectX 11 do jogo poderia travar na inicialização ou após carregar um jogo salvo em determinadas placas gráficas AMD.

          Adição de compatibilidade com Intel Xe Super Sampling, uma tecnologia de dimensionamento de resolução que usa aprendizado automático para melhorar o desempenho e a qualidade da imagem.

          Correção de alguns problemas de iluminação com Reflexos de Espaço de Tela quando a iluminação global com traçado de raios está direcionada.

          Melhorias de desempenho para iluminação global com traçado de raios e reflexos com traçado de raios.

          Correção de um problema em que respingos roxos podiam aparecer na tela ao passar montado no Carpeado por poças d'água.</p>

        <h2>Call Of Duty: Modern Warfare </h2>

        <br />

        <h3>Review Call of Duty Modern Warfare 3: jogo agrada, mas poderia ser DLC</h3>

        <br />

        <p>Call of Duty: Modern Warfare 3 é o novo game da série de FPS da Activision, disponível para PlayStation 5 (PS5), PlayStation 4 (PS4), Xbox Series X, Xbox Series S, Xbox One e PC (via Steam e Battle.net) por R$ 299. Disponível desde o dia 10 novembro de 2023, o jogo fecha a trilogia mais recente de Modern Warfare, mas traz poucas novidades em relação ao game anterior. Por isso, muitos usuários comentaram que o título, na realidade, está mais próximo de ser uma DLC do que um lançamento de fato.</p>

        <p>Ainda assim, MW3 traz uma gameplay frenética e que relembra as melhores fases da franquia. Apesar do seu preço cheio e história curta - são cerca de cinco horas, segundo o site HowLongToBeat - o multiplayer do novo Call of Duty é completo e traz 16 mapas clássicos. O modo Zumbis também é destaque, já que é a primeira vez que a saga Modern Warfare recebe a modalidade que é sucesso entre os fãs do FPS.</p>

      </div>

    </section>

  )

}

Sobre:

import NavBar from "@/components/navbar";

import Link from "next/link";

export default function Sobre() {

  return (

    <section>

      <NavBar />

      <div className="sobre">

        <h1 className="sobre-h1">Quem somos nós?</h1>

        <p className="sobre-p" >Somos 5 universitários que temos como objetivo levar entretenimento para os usuários por meio deste site. Estamos sempre em busca de conhecimento sobre o assunto, portanto, nosso objetivo é sempre estar aprendendo. Temos como sonho desenvolver, não só pelo salário ser ótimo (vale lembrar, haha) mas também pelo fato de sermos apaixonados pela área de TI. Gostamos muito de jogos, é ai que entra o questionamento.. Será que seremos não apenas desenvolvedores de sites, mas também desenvolvedores de jogos? Acompanhe os próximos capitulos ;). </p>

        <h2> Siga nossas redes sociais</h2>

        <div className="icons" >

          <Link href="#"><img src="/img/x.png" /></Link>

          <Link href="#"><img src="/img/discord.png" /></Link>

          <Link href="#"><img src="/img/instagram.png" /></Link>

          <Link href="#"><img src="/img/facebook.png" /></Link>

          <Link href="#"><img src="/img/github.png" /></Link>

        </div>

      </div>

    </section>

  )

}

Layout:

import { Inter } from 'next/font/google'

import 'bootstrap/dist/css/bootstrap.min.css'

import './globals.css'

const inter = Inter({ subsets: ['latin'] })

export const metadata = {

  title: 'Dev Game App',

  description: 'Generated by create next app',

}

export default function RootLayout({ children }) {

  return (

    <html lang="pt-br">

      <body className={inter.className}>{children}</body>

    </html>

  )

}

Page:

import Carrossel from '@/components/Carrossel';

import NavBar from '@/components/navbar';

export default function Home() {

  return (

    <main>

      <NavBar />

      <Carrossel />

    </main>

  )

}

**Componentes:**

CadastraJogo:

'use client'

import { createJogoAxios, getPlataformaAxios, updateGamesAxios } from "@/util/axios";

import { useEffect, useState } from "react"

export function CadastrarJogo() {

  const [platform, setPlatform] = useState([]);

  const [updateGame, setUpdateGame] = useState(null);

  const [nome, setNome] = useState('');

  const [cat, setCat] = useState('');

  const [plataforma\_que, setPlataforma\_que] = useState('');

  const [nota, setNota] = useState('');

  const [status, setStatus] = useState('');

  const [recomendacao, setRecomendacao] = useState('Sim');

  useEffect(() => {

    const fetchData = async () => {

      try {

        const result = await getPlataformaAxios();

        setPlatform(result);

        setUpdateGame(JSON.parse(localStorage.getItem('game')));

      } catch (error) {

        console.error('Error fetching data:', error);

      }

    };

    fetchData();

  }, []);

  useEffect(() => {

    const setInfo = () => {

      setNome(updateGame.nome)

      setCat(updateGame.cat)

      setPlataforma\_que(updateGame.plataforma\_disp)

      setNota(updateGame.nota)

      setStatus(updateGame.status)

      setRecomendacao(updateGame.recomendacao)

    }

    if (updateGame) setInfo()

  }, [updateGame]);

  const limparinputi = () => {

    setNome('')

    setCat('')

    setPlataforma\_que('')

    setNota('')

    setStatus('')

    setRecomendacao('')

  }

  const handleSubmit = async (e) => {

    e.preventDefault();

    try {

      const newJogo = { nome, cat, plataforma\_que, nota, status, recomendacao }

      const result = await createJogoAxios(newJogo)

      if (result) {

        limparinputi();

        alert(result);

      }

    } catch (error) {

      alert(error.response.data.message)

    }

  };

  const newUpdateGame = async (e) => {

    e.preventDefault();

    try {

      const newJogo = { nome, cat, plataforma\_que, nota, status, recomendacao }

      const result = await updateGamesAxios(updateGame.id, newJogo)

      if (result) {

        limparinputi();

        alert(result.message);

      }

      limparinputi();

      localStorage.removeItem('game');

    } catch (error) {

      alert(error.response.data.message)

    }

  }

  return (

    <form className="cj-form">

      <input

        type="text"

        value={nome}

        className="cj-input"

        onChange={(e) => setNome(e.target.value)}

        placeholder="Digite o nome do jogo"

      />

      <input

        type="text"

        value={cat}

        className="cj-input"

        onChange={(e) => setCat(e.target.value)}

        placeholder="Digite a categoria"

      />

      <select

        name="plataforma"

        className="cj-select select-cj"

        value={plataforma\_que}

        onChange={(e) => setPlataforma\_que(e.target.value)}

      >

        <option>Selecione Plataforma</option>

        {platform.map(({ id, nome }) => (

          <option key={id} value={nome}>

            {nome}

          </option>

        ))}

      </select>

      <input

        type="text"

        value={nota}

        className="cj-input"

        onChange={(e) => setNota(e.target.value)}

        placeholder="Digite uma nota"

      />

      <select

        name="status"

        value={status}

        className="cj-select select-cj"

        onChange={(e) => setStatus(e.target.value)}

      >

        <option>Selecione Status</option>

        <option value="Jogando">Jogando</option>

        <option value="Jogado">Jogado</option>

        <option value="Zerado">Zerado</option>

        <option value="Outros">Outros</option>

      </select>

      <select

        name="recomendacao"

        value={recomendacao}

        className="cj-select select-cj"

        onChange={(e) => setRecomendacao(e.target.value)}

      >

        <option value="Sim">Sim</option>

        <option value="Não">Não</option>

      </select>

      {

        updateGame ? <input

          type="submit"

          value="Atualizar"

          name="Atualizar"

          className="cj-input-btn"

          onClick={newUpdateGame}

        /> :

          <input

            type="submit"

            value="Enviar"

            name="Enviar"

            className="cj-input-btn"

            onClick={handleSubmit}

          />

      }

    </form>

  );

}

CadastraPlataforma:

'use client'

import { addJogoPlataformaAxios, createPlataformaAxios, deletJogosPlataformaAxios, gamesAxios, getJogosPlataformaAxios, getPlataformaAxios, updatePlataformaAxios } from "@/util/axios";

import { useEffect, useState } from "react";

export function CadastrarPlataforma() {

  const [updatePlataforma, setUpdatePlataforma] = useState(null);

  const [nome, setNome] = useState('');

  const [plataformaJogo, setPlataformaJogo] = useState([])

  const [jogoPlataforma, setJogoPlataforma] = useState([])

  const [idPlatform, getIdPlataforma] = useState('')

  const [idGame, getIdJogo] = useState('')

  const [listGame, setListGame] = useState([])

  const [render, setRender] = useState(false)

  useEffect(() => {

    const fetchData = async () => {

      const games = await gamesAxios()

      const platform = await getPlataformaAxios()

      setJogoPlataforma(games)

      setPlataformaJogo(platform)

    };

    fetchData();

  }, [])

  useEffect(() => {

    const fetchData = async () => {

      setUpdatePlataforma(JSON.parse(localStorage.getItem('plataforma')));

    };

    fetchData();

  }, [])

  useEffect(() => {

    const setInfo = () => {

      setNome(updatePlataforma.nome)

    }

    if (updatePlataforma) setInfo()

  }, [updatePlataforma]);

  const handleSubmit = async (e) => {

    e.preventDefault();

    try {

      const result = await createPlataformaAxios({ nome })

      setUpdatePlataforma(JSON.parse(localStorage.getItem('plataforma')));

      if (result) alert(result)

    } catch (error) {

      alert(error.response.data.message)

    }

  };

  const newUpdatePlataforma = async (e) => {

    e.preventDefault();

    try {

      if (nome === '') return alert('Preencha o campo nome')

      const result = await updatePlataformaAxios(updatePlataforma.id, { nome })

      if (result) {

        setNome('');

        alert(result.message);

      }

      setNome('');

      localStorage.removeItem('plataforma');

    } catch (error) {

      alert(error.response.data.message)

    }

  }

  useEffect(() => {

    const fetchData = async () => {

      const listGamePlataforma = await getJogosPlataformaAxios(idPlatform)

      console.log(listGamePlataforma);

      setRender(false)

      setListGame(listGamePlataforma)

    };

    if (idPlatform) fetchData();

  }, [idPlatform])

  useEffect(() => {

    const fetchData = async () => {

      const game = listGame.find(g => g.id === Number(idGame));

      if (game) setRender(true)

    };

    if (idPlatform) fetchData();

  }, [idGame])

  const adicionarJogo = async () => {

    try {

      const result = await addJogoPlataformaAxios({ idGame, idPlatform })

      if (result) alert(result)

    } catch (error) {

      alert(error.response.data.message)

    }

  }

  const deletarJogo = async () => {

    console.log('Deletar jogo da plataforma');

    try {

      const result = await deletJogosPlataformaAxios({ idGame, idPlatform })

      if (result) alert(result)

    } catch (error) {

      alert(error.response.data.message)

    }

  }

  return (

    <section>

      <form className="cj-form">

        <input

          type="text"

          value={nome}

          className="cj-input"

          onChange={(e) => setNome(e.target.value)}

          placeholder="Digite o nome da plataforma"

        />

        {

          updatePlataforma ?

            <input

              type="submit"

              value="Atualizar"

              name="Atualizar"

              className="cj-input-btn"

              onClick={newUpdatePlataforma}

            />

            : <input

              type="submit"

              value="Enviar"

              name="Enviar"

              className="cj-input-btn"

              onClick={handleSubmit}

            />

        }

      </form>

      <div>

        <h1 className="lista-jogos-h1 margin-top-20px">

          Adicionar Jogo a Plataforma

        </h1>

        <form className="cj-form">

          <select

            name="plataforma"

            className="cj-select select-cj"

            value={idPlatform}

            onChange={(e) => getIdPlataforma(e.target.value)}

          >

            <option>Selecione Plataforma</option>

            {plataformaJogo.map(({ id, nome }) => (

              <option key={id} value={id}>

                {nome}

              </option>

            ))}

          </select>

          <select

            name="plataforma"

            className="cj-select select-cj"

            value={idGame}

            onChange={(e) => getIdJogo(e.target.value)}

          >

            <option>Selecione Jogo</option>

            {jogoPlataforma.map(({ id, nome }) => (

              <option key={id} value={id}>

                {nome}

              </option>

            ))}

          </select>

          {

            render ?

              <input

                type="button"

                value="Deletar"

                className="cj-input-btn"

                onClick={deletarJogo}

              /> :

              <input

                type="button"

                value="Adicionar"

                className="cj-input-btn"

                onClick={adicionarJogo}

              />

          }

        </form>

      </div>

      <ul className="lista-jogos-ul">

        {

          listGame && listGame.map((game) => (

            <li key={game.id} className="lista-jogos-li">

              <p className="lista-jogos-p">{game.nome}</p>

            </li>

          ))

        }

      </ul>

    </section>

  )

}

Card:

import { addGamesAxios } from "@/util/axios";

export function Card({ userId, id, cat, nome, nota, plataforma\_disp, recomendacao, status }) {

  const addGames = async () => {

    console.log('Clicado');

    try {

      const response = await addGamesAxios({ userId, jogoId: id });

      if (response) alert(response);

    } catch (error) {

      alert(error.response.data.message)

    }

  };

  return (

    <div className="card-div">

      <h3 className="card-h3"> {nome}</h3>

      <p className="card-p"><span className="span\_card">Categoria:</span>{cat}</p>

      <p className="card-p"><span className="span\_card">Nota:</span>{nota}</p>

      <p className="card-p"><span className="span\_card">Plataforma Disp:</span>{plataforma\_disp}</p>

      <p className="card-p"><span className="span\_card">Recomendação:</span>{recomendacao}</p>

      <p className="card-p"><span className="span\_card">Status:</span>{status}</p>

      <button className="card-button" onClick={addGames}>

        Adicionar

      </button>

    </div>

  );

}

CardBiblioteca:

import { removeGamesAxios } from "@/util/axios";

export function CardBiblioteca({ userId, id, cat, nome, nota, plataforma\_disp, recomendacao, status }) {

  const removeGames = async () => {

    try {

      const response = await removeGamesAxios({ userId, jogoId: id });

      if (response) alert(response);

    } catch (error) {

      alert(error.response.data.message)

    }

  };

  return (

    <div className="card-div">

      <h3 className="card-h3">{nome}</h3>

      <p className="card-p">{cat}</p>

      <p className="card-p">{nota}</p>

      <p className="card-p">{plataforma\_disp}</p>

      <p className="card-p">{recomendacao}</p>

      <p className="card-p">{status}</p>

      <button className="card-button" onClick={removeGames}>

        Remover

      </button>

    </div>

  );

}

Carrossel:

'use client'

import { useEffect, useState } from 'react';

import Link from 'next/link';

export default function Carrossel() {

  const [cssActive\_1, setCssActive\_1] = useState('active')

  const [cssActive\_2, setCssActive\_2] = useState('')

  const [cssActive\_3, setCssActive\_3] = useState('')

  useEffect(() => {

    setTimeout(() => {

      if (cssActive\_1 === 'active') {

        setCssActive\_2('active')

        setCssActive\_1('')

      }

      if (cssActive\_2 === 'active') {

        setCssActive\_3('active')

        setCssActive\_2('')

      }

      if (cssActive\_3 === 'active') {

        setCssActive\_1('active')

        setCssActive\_3('')

      }

    }, 7000)

  }, [cssActive\_1, cssActive\_2, cssActive\_3])

  const nextSlide = () => {

    if (cssActive\_1 === 'active') {

      setCssActive\_2('active')

      setCssActive\_1('')

    }

    if (cssActive\_2 === 'active') {

      setCssActive\_3('active')

      setCssActive\_2('')

    }

    if (cssActive\_3 === 'active') {

      setCssActive\_1('active')

      setCssActive\_3('')

    }

  }

  const prevSlide = () => {

    if (cssActive\_1 === 'active') {

      setCssActive\_3('active')

      setCssActive\_1('')

    }

    if (cssActive\_2 === 'active') {

      setCssActive\_1('active')

      setCssActive\_2('')

    }

    if (cssActive\_3 === 'active') {

      setCssActive\_2('active')

      setCssActive\_3('')

    }

  }

  return (

    <div id="myCarousel" className="carousel slide" data-bs-ride="carousel">

      <div className="carousel-inner">

        <div className={`carousel-item ${cssActive\_1}`}>

          <img src="/img/the-witcher-jogo.png" className="d-block w-100 carousel-img" alt="The Witcher" />

          <div className="carousel-caption d-none d-md-block">

            <h5 className="carousel-caption-h5">The Witcher</h5>

            <p className="carousel-caption-p">Você é Geralt de Rívia, mercenário matador de monstros. O lugar é um continente devastado pela guerra e infestado de monstros que você pode explorar à vontade. Seu contrato atual? Encontrar Ciri, a Criança da Profecia — uma arma viva que pode alterar a forma do mundo.</p>

            <Link className="btn btn-primary" target="\_blank" href="https://www.thewitcher.com/us/pt-br/">

              Jogar

            </Link>

          </div>

        </div>

        <div className={`carousel-item ${cssActive\_2}`}>

          <img src="/img/rdr2.png" className="d-block w-100 carousel-img" alt="Red Dead Redemption 2" />

          <div className="carousel-caption d-none d-md-block">

            <h5 className="carousel-caption-h5">Red Dead Redemption 2</h5>

            <p className="carousel-caption-p">É um jogo eletrônico de ação-aventura desenvolvido pela  Rockstar Studios  e publicado pela  Rockstar Games. É uma prequela  de  Red Dead Redemption  de 2010, sendo o terceiro título da franquia  Red Dead. O enredo centra-se no bandido Arthur Morgan, um dos principais membros da gangue Van der Linde.</p>

            <Link className="btn btn-primary" target="\_blank" href="https://www.rockstargames.com/reddeadredemption2/">

              Jogar

            </Link>

          </div>

        </div>

        <div className={`carousel-item ${cssActive\_3}`}>

          <img src="/img/cod.png" className="d-block w-100 carousel-img" alt="Call of Duty" />

          <div className="carousel-caption d-none d-md-block">

            <h5 className="carousel-caption-h5">CALL OF DUTY</h5>

            <p className="carousel-caption-p">Call of Duty é um jogo eletrônico de tiro em primeira pessoa desenvolvido pela Infinity Ward e publicado pela Activision. Foi lançado em 29 de outubro de 2003 para Microsoft Windows.</p>

            <Link className="btn btn-primary" target="\_blank" href="https://www.callofduty.com/br/pt">

              Jogar

            </Link>

          </div>

        </div>

      </div>

      <button

        className="carousel-control-prev"

        type="button"

        data-bs-target="#myCarousel"

        data-bs-slide="prev"

        onClick={prevSlide}

      >

        <span className="carousel-control-prev-icon" aria-hidden="true"></span>

        <span className="visually-hidden">Anterior</span>

      </button>

      <button

        className="carousel-control-next"

        type="button"

        data-bs-target="#myCarousel"

        data-bs-slide="next"

        onClick={nextSlide}

      >

        <span className="carousel-control-next-icon" aria-hidden="true"></span>

        <span className="visually-hidden">Próximo</span>

      </button>

    </div>

  );

};

ListaJogo:

'use client'

import { deletGamesAxios, gamesAxios } from "@/util/axios";

import { useEffect, useState } from "react";

import { CadastrarJogo } from "./cadastraJogo";

export function ListarJogos() {

  const [render, setRender] = useState(true)

  const [games, setGames] = useState([])

  useEffect(() => {

    const fetchData = async () => {

      try {

        const result = await gamesAxios();

        setGames(result);

      } catch (error) {

        console.error('Error fetching data:', error);

      }

    };

    fetchData();

  }, [games])

  const delitGame = async (id) => {

    try {

      const result = await deletGamesAxios(id);

      if (result) alert(result);

    } catch (error) {

      alert(error.response.data.message);

    }

  }

  const addUser = async () => {

    localStorage.removeItem('game')

    setRender(false)

  }

  const updateGame = (game) => {

    localStorage.setItem('game', JSON.stringify(game))

    setRender(false)

  }

  return (

    <section className="lista-jogos">

      <div className="lista-jogos-title">

        <h1 className="lista-jogos-h1" onClick={() => setRender(true)}>Lista de Jogos</h1>

        <img src="/img/botao-adicionar.png" onClick={addUser} width={35} alt="Adicionar usuario" />

      </div>

      {

        render ?

        <ul className="lista-jogos-ul">

        {

          games && games.map((game) => (

            <li key={game.id} className="lista-jogos-li">

              <p className="lista-jogos-p">{game.nome}</p>

              <span>

                <input className="lista-jogos-input" type="button" value="Editar" onClick={() => updateGame(game)} />

                <input className="lista-jogos-input" type="button" value="Excluir" onClick={() => delitGame(game.id)} />

              </span>

            </li>

          ))

        }

      </ul> :

        <CadastrarJogo />

      }

    </section>

  )

}

ListaPlataforma:

'use client'

import { deletPlataformaAxios, getPlataformaAxios } from "@/util/axios"

import { useEffect, useState } from "react"

import { CadastrarPlataforma } from "./cadastraPlataforma"

export function ListaPlataforma() {

  const [render, setRender] = useState(true)

  const [plataforma, setPlataforma] = useState([])

  useEffect(() => {

    const fetchData = async () => {

      try {

        const result = await getPlataformaAxios();

        setPlataforma(result);

      } catch (error) {

        console.error(error.response.data.message);

      }

    };

    fetchData();

  }, [plataforma])

  const addUser = async () => {

    localStorage.removeItem('plataforma')

    setRender(false)

  }

  const updatePlataforma = async (plataforma) => {

    localStorage.setItem('plataforma', JSON.stringify(plataforma))

    setRender(false)

  }

  const delitPlataforma = async (id) => {

    try {

      const result = await deletPlataformaAxios(id);

      if (result) alert(result);

    } catch (error) {

      alert(error.response.data.message);

    }

  }

  return (

    <section className="lista-jogos">

      <div className="lista-jogos-title">

        <h1 className="lista-jogos-h1" onClick={() => setRender(true)}>Lista</h1>

        <img src="/img/botao-adicionar.png" onClick={addUser} width={35} alt="Adicionar plataforma" />

      </div>

      {

        render ?

          <ul className="lista-jogos-ul">

            {

              plataforma && plataforma.map((plataforma) => (

                <li key={plataforma.id} className="lista-jogos-li">

                  <p className="lista-jogos-p">{plataforma.nome}</p>

                  <span>

                    <input className="lista-jogos-input" type="button" value="Editar" onClick={() => updatePlataforma(plataforma)} />

                    <input className="lista-jogos-input" type="button" value="Excluir" onClick={() => delitPlataforma(plataforma.id)} />

                  </span>

                </li>

              ))

            }

          </ul> :

          <CadastrarPlataforma />

      }

    </section>

  )

}

NavBar:

import Image from "next/image";

import Link from "next/link";

export default function NavBar() {

  return (

    <nav className="nav">

      <div className="nav-div nav-logo">

        <Link className="nav-logo-link" href="/">

          <Image src="/img/logo.png" width={40} height={40} alt="Logo" />

        </Link>

        <div className="nav-div">

          <ul className="nav-ul">

            <li className="nav-li nav-li-link:hover"><Link className="nav-li-link" href="/noticias">Noticias</Link></li>

            <li className="nav-li nav-li-link:hover"><Link className="nav-li-link" href="/comunidade">Comunidade</Link></li>

            <li className="nav-li nav-li-link:hover"><Link className="nav-li-link" href="/sobre">Sobre Nós</Link></li>

          </ul>

        </div>

      </div>

      <div className="nav-div">

        <ul className="nav-ul">

          <li className="nav-li nav-li-link:hover"><Link className="nav-li-link" href="/login">Login</Link></li>

          <li className="nav-li nav-li-link:hover"><Link className="nav-li-link" href="/createUser">Criar Conta</Link></li>

        </ul>

      </div>

    </nav>

  )

}

NavBarHome:

'use client'

import { deleteCookie } from "cookies-next";

import Image from "next/image";

import Link from "next/link";

export default function NavBarHome() {

  const logout = () => {

    localStorage.removeItem('user')

    deleteCookie('user')

  }

  return (

    <nav className="nav">

      <div className="nav-div nav-logo">

        <Link className="nav-logo-link" href="/">

          <Image src="/img/logo.png" width={40} height={40} alt="Logo" />

        </Link>

      </div>

      <div className="nav-div">

        <ul className="nav-ul">

          <li className="nav-li nav-li-link:hover">

            <Link className="nav-li-link" href="/home">Jogos</Link>

          </li>

          <li className="nav-li nav-li-link:hover">

            <Link className="nav-li-link" href="/cadastro">Cadastrar</Link>

          </li>

          <li className="nav-li nav-li-link:hover">

            <Link className="nav-li-link" href="/biblioteca">Biblioteca</Link>

          </li>

          <li className="nav-li nav-li-link:hover">

            <Link className="nav-li-link" href="/noticias">Noticias</Link>

          </li>

          <li className="nav-li nav-li-link:hover">

            <Link className="nav-li-link" href="/comunidade">Comunidade</Link>

          </li>

          <li className="nav-li nav-li-link:hover">

            <Link className="nav-li-link" href="/" onClick={logout}>Sair</Link>

          </li>

        </ul>

      </div>

    </nav>

  )

}

**Ulitidades**:

Axios:

import axios from 'axios'

const URL\_BASE = 'http://localhost:3001';

// Rota User

export const loginAxios = async (data) => {

  try {

    const URL = `${URL\_BASE}/user/login`;

    const response = await axios.post(URL, data);

    return response.data;

  } catch (error) {

    throw error;

  }

};

export const getUserAxios = async (data) => {

  try {

    const URL = `${URL\_BASE}/user`;

    const response = await axios.post(URL, data);

    return response.data;

  } catch (error) {

    throw error;

  }

};

export const addGamesAxios = async (data) => {

  try {

    const URL = `${URL\_BASE}/user/game`;

    const response = await axios.post(URL, data);

    return response.data.message;

  } catch (error) {

    throw error;

  }

};

export const getGamesAxios = async (id) => {

  try {

    const URL = `${URL\_BASE}/user/games/${id}`;

    const response = await axios.get(URL);

    return response.data;

  } catch (error) {

    throw error;

  }

};

export const updateUserAxios = async (id, data) => {

  try {

    const URL = `${URL\_BASE}/user/update/${id}`;

    const response = await axios.patch(URL, data);

    return response.data.message;

  } catch (error) {

    throw error;

  }

};

export const removeGamesAxios = async (data) => {

  try {

    const URL = `${URL\_BASE}/user/delete-game`;

    const response = await axios.post(URL, data);

    return response.data.message;

  } catch (error) {

    throw error;

  }

};

export const createUserAxios = async (data) => {

  try {

    const URL = `${URL\_BASE}/user/create`;

    const response = await axios.post(URL, data);

    return response.data.message;

  } catch (error) {

    throw error;

  }

};

// Rota Jogo

export const gamesAxios = async () => {

  try {

    const URL = `${URL\_BASE}/game`;

    const response = await axios.get(URL);

    return response.data;

  } catch (error) {

    throw error;

  }

};

export const createJogoAxios = async (data) => {

  try {

    const URL = `${URL\_BASE}/game/insert`;

    const response = await axios.post(URL, data);

    return response.data.message;

  } catch (error) {

    throw error;

  }

};

export const deletGamesAxios = async (id) => {

  try {

    const URL = `${URL\_BASE}/game/delete/${id}`;

    const response = await axios.delete(URL);

    return response.data.message;

  } catch (error) {

    throw error;

  }

};

export const updateGamesAxios = async (id, data) => {

  try {

    const URL = `${URL\_BASE}/game/update/${id}`;

    const response = await axios.patch(URL, data);

    return response.data;

  } catch (error) {

    throw error;

  }

};

// Rota Plataforma

export const getPlataformaAxios = async () => {

  try {

    const URL = `${URL\_BASE}/platform`;

    const response = await axios.get(URL);

    return response.data;

  } catch (error) {

    throw error;

  }

};

export const createPlataformaAxios = async (data) => {

  try {

    const URL = `${URL\_BASE}/platform/insert`;

    const response = await axios.post(URL, data);

    return response.data.message;

  } catch (error) {

    throw error;

  }

};

export const updatePlataformaAxios = async (id, data) => {

  try {

    const URL = `${URL\_BASE}/platform/update/${id}`;

    const response = await axios.patch(URL, data);

    return response.data;

  } catch (error) {

    throw error;

  }

};

export const deletPlataformaAxios = async (id) => {

  try {

    const URL = `${URL\_BASE}/platform/delete/${id}`;

    const response = await axios.delete(URL);

    return response.data.message;

  } catch (error) {

    throw error;

  }

};

export const addJogoPlataformaAxios = async (data) => {

  try {

    const URL = `${URL\_BASE}/platform/game`;

    const response = await axios.post(URL, data);

    return response.data.message;

  } catch (error) {

    throw error;

  }

};

export const getJogosPlataformaAxios = async (id) => {

  try {

    const URL = `${URL\_BASE}/platform/${id}/games`;

    const response = await axios.get(URL);

    return response.data;

  } catch (error) {

    throw error;

  }

};

export const deletJogosPlataformaAxios = async (data) => {

  try {

    const URL = `${URL\_BASE}/platform/delete-game`;

    const response = await axios.delete(URL, {data});

    return response.data.message;

  } catch (error) {

    throw error;

  }

};

SendEmail:

'use server'

const nodemailer = require('nodemailer');

export async function sendEmail(email, userName) {

  // Configura o transporte do e-mail (altere com suas próprias credenciais)

  const transporter = nodemailer.createTransport({

    service: 'gmail',

    auth: {

      user: 'devgameunifacs@gmail.com',

      pass: 'xygi llqy nriw crnd',

    },

  });

  // Configura o conteúdo do e-mail

  const mailOptions = {

    from: 'devgameunifacs@gmail.com',

    to: email,

    subject: 'Recuperar Senha',

    text: `

      Recuperação de senha do Dev Games

      Clique no link abaixo para alterar sua senha.

      http://localhost:3000/esqueceuSenha/${userName}

    `,

  };

  // Tenta enviar o e-mail

  try {

    const info = await transporter.sendMail(mailOptions);

    return info.messageId

  } catch (error) {

    console.error('Erro ao enviar o e-mail:', error);

  }

}

Middleware:

import { NextResponse } from 'next/server'

export async function middleware(request) {

  const user = request.cookies.get('user')?.value

  if (!user) {

    if (request.nextUrl.pathname === '/') return NextResponse.next()

    return NextResponse.redirect(new URL('/', request.url))

  }

}

export const config = {

  matcher: ['/', '/home', '/user/:path\*', '/platform/:path\*', '/game/:path\*', '/biblioteca/:path\*'],

}

**CONFIG**

Package:

{

  "name": "front-end",

  "version": "0.1.0",

  "private": true,

  "scripts": {

    "dev": "next dev",

    "build": "next build",

    "start": "next start",

    "lint": "next lint"

  },

  "dependencies": {

    "axios": "^1.6.2",

    "bootstrap": "^5.3.2",

    "cookies-next": "^4.1.0",

    "next": "14.0.3",

    "nodemailer": "^6.9.7",

    "react": "^18",

    "react-dom": "^18"

  }

}

Package-lock:

{

  "name": "front-end",

  "version": "0.1.0",

  "lockfileVersion": 3,

  "requires": true,

  "packages": {

    "": {

      "name": "front-end",

      "version": "0.1.0",

      "dependencies": {

        "axios": "^1.6.2",

        "bootstrap": "^5.3.2",

        "cookies-next": "^4.1.0",

        "next": "14.0.3",

        "nodemailer": "^6.9.7",

        "react": "^18",

        "react-dom": "^18"

      }

    },

    "node\_modules/@next/env": {

      "version": "14.0.3",

      "resolved": "https://registry.npmjs.org/@next/env/-/env-14.0.3.tgz",

      "integrity": "sha512-7xRqh9nMvP5xrW4/+L0jgRRX+HoNRGnfJpD+5Wq6/13j3dsdzxO3BCXn7D3hMqsDb+vjZnJq+vI7+EtgrYZTeA=="

    },

    "node\_modules/@next/swc-darwin-arm64": {

      "version": "14.0.3",

      "resolved": "https://registry.npmjs.org/@next/swc-darwin-arm64/-/swc-darwin-arm64-14.0.3.tgz",

      "integrity": "sha512-64JbSvi3nbbcEtyitNn2LEDS/hcleAFpHdykpcnrstITFlzFgB/bW0ER5/SJJwUPj+ZPY+z3e+1jAfcczRLVGw==",

      "cpu": [

        "arm64"

      ],

      "optional": true,

      "os": [

        "darwin"

      ],

      "engines": {

        "node": ">= 10"

      }

    },

    "node\_modules/@next/swc-darwin-x64": {

      "version": "14.0.3",

      "resolved": "https://registry.npmjs.org/@next/swc-darwin-x64/-/swc-darwin-x64-14.0.3.tgz",

      "integrity": "sha512-RkTf+KbAD0SgYdVn1XzqE/+sIxYGB7NLMZRn9I4Z24afrhUpVJx6L8hsRnIwxz3ERE2NFURNliPjJ2QNfnWicQ==",

      "cpu": [

        "x64"

      ],

      "optional": true,

      "os": [

        "darwin"

      ],

      "engines": {

        "node": ">= 10"

      }

    },

    "node\_modules/@next/swc-linux-arm64-gnu": {

      "version": "14.0.3",

      "resolved": "https://registry.npmjs.org/@next/swc-linux-arm64-gnu/-/swc-linux-arm64-gnu-14.0.3.tgz",

      "integrity": "sha512-3tBWGgz7M9RKLO6sPWC6c4pAw4geujSwQ7q7Si4d6bo0l6cLs4tmO+lnSwFp1Tm3lxwfMk0SgkJT7EdwYSJvcg==",

      "cpu": [

        "arm64"

      ],

      "optional": true,

      "os": [

        "linux"

      ],

      "engines": {

        "node": ">= 10"

      }

    },

    "node\_modules/@next/swc-linux-arm64-musl": {

      "version": "14.0.3",

      "resolved": "https://registry.npmjs.org/@next/swc-linux-arm64-musl/-/swc-linux-arm64-musl-14.0.3.tgz",

      "integrity": "sha512-v0v8Kb8j8T23jvVUWZeA2D8+izWspeyeDGNaT2/mTHWp7+37fiNfL8bmBWiOmeumXkacM/AB0XOUQvEbncSnHA==",

      "cpu": [

        "arm64"

      ],

      "optional": true,

      "os": [

        "linux"

      ],

      "engines": {

        "node": ">= 10"

      }

    },

    "node\_modules/@next/swc-linux-x64-gnu": {

      "version": "14.0.3",

      "resolved": "https://registry.npmjs.org/@next/swc-linux-x64-gnu/-/swc-linux-x64-gnu-14.0.3.tgz",

      "integrity": "sha512-VM1aE1tJKLBwMGtyBR21yy+STfl0MapMQnNrXkxeyLs0GFv/kZqXS5Jw/TQ3TSUnbv0QPDf/X8sDXuMtSgG6eg==",

      "cpu": [

        "x64"

      ],

      "optional": true,

      "os": [

        "linux"

      ],

      "engines": {

        "node": ">= 10"

      }

    },

    "node\_modules/@next/swc-linux-x64-musl": {

      "version": "14.0.3",

      "resolved": "https://registry.npmjs.org/@next/swc-linux-x64-musl/-/swc-linux-x64-musl-14.0.3.tgz",

      "integrity": "sha512-64EnmKy18MYFL5CzLaSuUn561hbO1Gk16jM/KHznYP3iCIfF9e3yULtHaMy0D8zbHfxset9LTOv6cuYKJgcOxg==",

      "cpu": [

        "x64"

      ],

      "optional": true,

      "os": [

        "linux"

      ],

      "engines": {

        "node": ">= 10"

      }

    },

    "node\_modules/@next/swc-win32-arm64-msvc": {

      "version": "14.0.3",

      "resolved": "https://registry.npmjs.org/@next/swc-win32-arm64-msvc/-/swc-win32-arm64-msvc-14.0.3.tgz",

      "integrity": "sha512-WRDp8QrmsL1bbGtsh5GqQ/KWulmrnMBgbnb+59qNTW1kVi1nG/2ndZLkcbs2GX7NpFLlToLRMWSQXmPzQm4tog==",

      "cpu": [

        "arm64"

      ],

      "optional": true,

      "os": [

        "win32"

      ],

      "engines": {

        "node": ">= 10"

      }

    },

    "node\_modules/@next/swc-win32-ia32-msvc": {

      "version": "14.0.3",

      "resolved": "https://registry.npmjs.org/@next/swc-win32-ia32-msvc/-/swc-win32-ia32-msvc-14.0.3.tgz",

      "integrity": "sha512-EKffQeqCrj+t6qFFhIFTRoqb2QwX1mU7iTOvMyLbYw3QtqTw9sMwjykyiMlZlrfm2a4fA84+/aeW+PMg1MjuTg==",

      "cpu": [

        "ia32"

      ],

      "optional": true,

      "os": [

        "win32"

      ],

      "engines": {

        "node": ">= 10"

      }

    },

    "node\_modules/@next/swc-win32-x64-msvc": {

      "version": "14.0.3",

      "resolved": "https://registry.npmjs.org/@next/swc-win32-x64-msvc/-/swc-win32-x64-msvc-14.0.3.tgz",

      "integrity": "sha512-ERhKPSJ1vQrPiwrs15Pjz/rvDHZmkmvbf/BjPN/UCOI++ODftT0GtasDPi0j+y6PPJi5HsXw+dpRaXUaw4vjuQ==",

      "cpu": [

        "x64"

      ],

      "optional": true,

      "os": [

        "win32"

      ],

      "engines": {

        "node": ">= 10"

      }

    },

    "node\_modules/@popperjs/core": {

      "version": "2.11.8",

      "resolved": "https://registry.npmjs.org/@popperjs/core/-/core-2.11.8.tgz",

      "integrity": "sha512-P1st0aksCrn9sGZhp8GMYwBnQsbvAWsZAX44oXNNvLHGqAOcoVxmjZiohstwQ7SqKnbR47akdNi+uleWD8+g6A==",

      "peer": true,

      "funding": {

        "type": "opencollective",

        "url": "https://opencollective.com/popperjs"

      }

    },

    "node\_modules/@swc/helpers": {

      "version": "0.5.2",

      "resolved": "https://registry.npmjs.org/@swc/helpers/-/helpers-0.5.2.tgz",

      "integrity": "sha512-E4KcWTpoLHqwPHLxidpOqQbcrZVgi0rsmmZXUle1jXmJfuIf/UWpczUJ7MZZ5tlxytgJXyp0w4PGkkeLiuIdZw==",

      "dependencies": {

        "tslib": "^2.4.0"

      }

    },

    "node\_modules/@types/cookie": {

      "version": "0.4.1",

      "resolved": "https://registry.npmjs.org/@types/cookie/-/cookie-0.4.1.tgz",

      "integrity": "sha512-XW/Aa8APYr6jSVVA1y/DEIZX0/GMKLEVekNG727R8cs56ahETkRAy/3DR7+fJyh7oUgGwNQaRfXCun0+KbWY7Q=="

    },

    "node\_modules/@types/node": {

      "version": "16.18.66",

      "resolved": "https://registry.npmjs.org/@types/node/-/node-16.18.66.tgz",

      "integrity": "sha512-sePmD/imfKvC4re/Wwos1NEcXYm6O96CAG5gQVY53nmDb8ePQ4qPku6uruN7n6TJ0t5FhcoUc2+yvE2/UZVDZw=="

    },

    "node\_modules/asynckit": {

      "version": "0.4.0",

      "resolved": "https://registry.npmjs.org/asynckit/-/asynckit-0.4.0.tgz",

      "integrity": "sha512-Oei9OH4tRh0YqU3GxhX79dM/mwVgvbZJaSNaRk+bshkj0S5cfHcgYakreBjrHwatXKbz+IoIdYLxrKim2MjW0Q=="

    },

    "node\_modules/axios": {

      "version": "1.6.2",

      "resolved": "https://registry.npmjs.org/axios/-/axios-1.6.2.tgz",

      "integrity": "sha512-7i24Ri4pmDRfJTR7LDBhsOTtcm+9kjX5WiY1X3wIisx6G9So3pfMkEiU7emUBe46oceVImccTEM3k6C5dbVW8A==",

      "dependencies": {

        "follow-redirects": "^1.15.0",

        "form-data": "^4.0.0",

        "proxy-from-env": "^1.1.0"

      }

    },

    "node\_modules/bootstrap": {

      "version": "5.3.2",

      "resolved": "https://registry.npmjs.org/bootstrap/-/bootstrap-5.3.2.tgz",

      "integrity": "sha512-D32nmNWiQHo94BKHLmOrdjlL05q1c8oxbtBphQFb9Z5to6eGRDCm0QgeaZ4zFBHzfg2++rqa2JkqCcxDy0sH0g==",

      "funding": [

        {

          "type": "github",

          "url": "https://github.com/sponsors/twbs"

        },

        {

          "type": "opencollective",

          "url": "https://opencollective.com/bootstrap"

        }

      ],

      "peerDependencies": {

        "@popperjs/core": "^2.11.8"

      }

    },

    "node\_modules/busboy": {

      "version": "1.6.0",

      "resolved": "https://registry.npmjs.org/busboy/-/busboy-1.6.0.tgz",

      "integrity": "sha512-8SFQbg/0hQ9xy3UNTB0YEnsNBbWfhf7RtnzpL7TkBiTBRfrQ9Fxcnz7VJsleJpyp6rVLvXiuORqjlHi5q+PYuA==",

      "dependencies": {

        "streamsearch": "^1.1.0"

      },

      "engines": {

        "node": ">=10.16.0"

      }

    },

    "node\_modules/caniuse-lite": {

      "version": "1.0.30001565",

      "resolved": "https://registry.npmjs.org/caniuse-lite/-/caniuse-lite-1.0.30001565.tgz",

      "integrity": "sha512-xrE//a3O7TP0vaJ8ikzkD2c2NgcVUvsEe2IvFTntV4Yd1Z9FVzh+gW+enX96L0psrbaFMcVcH2l90xNuGDWc8w==",

      "funding": [

        {

          "type": "opencollective",

          "url": "https://opencollective.com/browserslist"

        },

        {

          "type": "tidelift",

          "url": "https://tidelift.com/funding/github/npm/caniuse-lite"

        },

        {

          "type": "github",

          "url": "https://github.com/sponsors/ai"

        }

      ]

    },

    "node\_modules/client-only": {

      "version": "0.0.1",

      "resolved": "https://registry.npmjs.org/client-only/-/client-only-0.0.1.tgz",

      "integrity": "sha512-IV3Ou0jSMzZrd3pZ48nLkT9DA7Ag1pnPzaiQhpW7c3RbcqqzvzzVu+L8gfqMp/8IM2MQtSiqaCxrrcfu8I8rMA=="

    },

    "node\_modules/combined-stream": {

      "version": "1.0.8",

      "resolved": "https://registry.npmjs.org/combined-stream/-/combined-stream-1.0.8.tgz",

      "integrity": "sha512-FQN4MRfuJeHf7cBbBMJFXhKSDq+2kAArBlmRBvcvFE5BB1HZKXtSFASDhdlz9zOYwxh8lDdnvmMOe/+5cdoEdg==",

      "dependencies": {

        "delayed-stream": "~1.0.0"

      },

      "engines": {

        "node": ">= 0.8"

      }

    },

    "node\_modules/cookie": {

      "version": "0.4.2",

      "resolved": "https://registry.npmjs.org/cookie/-/cookie-0.4.2.tgz",

      "integrity": "sha512-aSWTXFzaKWkvHO1Ny/s+ePFpvKsPnjc551iI41v3ny/ow6tBG5Vd+FuqGNhh1LxOmVzOlGUriIlOaokOvhaStA==",

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/cookies-next": {

      "version": "4.1.0",

      "resolved": "https://registry.npmjs.org/cookies-next/-/cookies-next-4.1.0.tgz",

      "integrity": "sha512-BREVc4TJT4NwXfyKjdjnYFXM6iRns+MYpCd34ClXuYqeisXnkPkbq7Ok9xaqi9mHmV6H2rwPE+p3EpMz4pF/kQ==",

      "dependencies": {

        "@types/cookie": "^0.4.1",

        "@types/node": "^16.10.2",

        "cookie": "^0.4.0"

      }

    },

    "node\_modules/delayed-stream": {

      "version": "1.0.0",

      "resolved": "https://registry.npmjs.org/delayed-stream/-/delayed-stream-1.0.0.tgz",

      "integrity": "sha512-ZySD7Nf91aLB0RxL4KGrKHBXl7Eds1DAmEdcoVawXnLD7SDhpNgtuII2aAkg7a7QS41jxPSZ17p4VdGnMHk3MQ==",

      "engines": {

        "node": ">=0.4.0"

      }

    },

    "node\_modules/follow-redirects": {

      "version": "1.15.3",

      "resolved": "https://registry.npmjs.org/follow-redirects/-/follow-redirects-1.15.3.tgz",

      "integrity": "sha512-1VzOtuEM8pC9SFU1E+8KfTjZyMztRsgEfwQl44z8A25uy13jSzTj6dyK2Df52iV0vgHCfBwLhDWevLn95w5v6Q==",

      "funding": [

        {

          "type": "individual",

          "url": "https://github.com/sponsors/RubenVerborgh"

        }

      ],

      "engines": {

        "node": ">=4.0"

      },

      "peerDependenciesMeta": {

        "debug": {

          "optional": true

        }

      }

    },

    "node\_modules/form-data": {

      "version": "4.0.0",

      "resolved": "https://registry.npmjs.org/form-data/-/form-data-4.0.0.tgz",

      "integrity": "sha512-ETEklSGi5t0QMZuiXoA/Q6vcnxcLQP5vdugSpuAyi6SVGi2clPPp+xgEhuMaHC+zGgn31Kd235W35f7Hykkaww==",

      "dependencies": {

        "asynckit": "^0.4.0",

        "combined-stream": "^1.0.8",

        "mime-types": "^2.1.12"

      },

      "engines": {

        "node": ">= 6"

      }

    },

    "node\_modules/glob-to-regexp": {

      "version": "0.4.1",

      "resolved": "https://registry.npmjs.org/glob-to-regexp/-/glob-to-regexp-0.4.1.tgz",

      "integrity": "sha512-lkX1HJXwyMcprw/5YUZc2s7DrpAiHB21/V+E1rHUrVNokkvB6bqMzT0VfV6/86ZNabt1k14YOIaT7nDvOX3Iiw=="

    },

    "node\_modules/graceful-fs": {

      "version": "4.2.11",

      "resolved": "https://registry.npmjs.org/graceful-fs/-/graceful-fs-4.2.11.tgz",

      "integrity": "sha512-RbJ5/jmFcNNCcDV5o9eTnBLJ/HszWV0P73bc+Ff4nS/rJj+YaS6IGyiOL0VoBYX+l1Wrl3k63h/KrH+nhJ0XvQ=="

    },

    "node\_modules/js-tokens": {

      "version": "4.0.0",

      "resolved": "https://registry.npmjs.org/js-tokens/-/js-tokens-4.0.0.tgz",

      "integrity": "sha512-RdJUflcE3cUzKiMqQgsCu06FPu9UdIJO0beYbPhHN4k6apgJtifcoCtT9bcxOpYBtpD2kCM6Sbzg4CausW/PKQ=="

    },

    "node\_modules/loose-envify": {

      "version": "1.4.0",

      "resolved": "https://registry.npmjs.org/loose-envify/-/loose-envify-1.4.0.tgz",

      "integrity": "sha512-lyuxPGr/Wfhrlem2CL/UcnUc1zcqKAImBDzukY7Y5F/yQiNdko6+fRLevlw1HgMySw7f611UIY408EtxRSoK3Q==",

      "dependencies": {

        "js-tokens": "^3.0.0 || ^4.0.0"

      },

      "bin": {

        "loose-envify": "cli.js"

      }

    },

    "node\_modules/mime-db": {

      "version": "1.52.0",

      "resolved": "https://registry.npmjs.org/mime-db/-/mime-db-1.52.0.tgz",

      "integrity": "sha512-sPU4uV7dYlvtWJxwwxHD0PuihVNiE7TyAbQ5SWxDCB9mUYvOgroQOwYQQOKPJ8CIbE+1ETVlOoK1UC2nU3gYvg==",

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/mime-types": {

      "version": "2.1.35",

      "resolved": "https://registry.npmjs.org/mime-types/-/mime-types-2.1.35.tgz",

      "integrity": "sha512-ZDY+bPm5zTTF+YpCrAU9nK0UgICYPT0QtT1NZWFv4s++TNkcgVaT0g6+4R2uI4MjQjzysHB1zxuWL50hzaeXiw==",

      "dependencies": {

        "mime-db": "1.52.0"

      },

      "engines": {

        "node": ">= 0.6"

      }

    },

    "node\_modules/nanoid": {

      "version": "3.3.7",

      "resolved": "https://registry.npmjs.org/nanoid/-/nanoid-3.3.7.tgz",

      "integrity": "sha512-eSRppjcPIatRIMC1U6UngP8XFcz8MQWGQdt1MTBQ7NaAmvXDfvNxbvWV3x2y6CdEUciCSsDHDQZbhYaB8QEo2g==",

      "funding": [

        {

          "type": "github",

          "url": "https://github.com/sponsors/ai"

        }

      ],

      "bin": {

        "nanoid": "bin/nanoid.cjs"

      },

      "engines": {

        "node": "^10 || ^12 || ^13.7 || ^14 || >=15.0.1"

      }

    },

    "node\_modules/next": {

      "version": "14.0.3",

      "resolved": "https://registry.npmjs.org/next/-/next-14.0.3.tgz",

      "integrity": "sha512-AbYdRNfImBr3XGtvnwOxq8ekVCwbFTv/UJoLwmaX89nk9i051AEY4/HAWzU0YpaTDw8IofUpmuIlvzWF13jxIw==",

      "dependencies": {

        "@next/env": "14.0.3",

        "@swc/helpers": "0.5.2",

        "busboy": "1.6.0",

        "caniuse-lite": "^1.0.30001406",

        "postcss": "8.4.31",

        "styled-jsx": "5.1.1",

        "watchpack": "2.4.0"

      },

      "bin": {

        "next": "dist/bin/next"

      },

      "engines": {

        "node": ">=18.17.0"

      },

      "optionalDependencies": {

        "@next/swc-darwin-arm64": "14.0.3",

        "@next/swc-darwin-x64": "14.0.3",

        "@next/swc-linux-arm64-gnu": "14.0.3",

        "@next/swc-linux-arm64-musl": "14.0.3",

        "@next/swc-linux-x64-gnu": "14.0.3",

        "@next/swc-linux-x64-musl": "14.0.3",

        "@next/swc-win32-arm64-msvc": "14.0.3",

        "@next/swc-win32-ia32-msvc": "14.0.3",

        "@next/swc-win32-x64-msvc": "14.0.3"

      },

      "peerDependencies": {

        "@opentelemetry/api": "^1.1.0",

        "react": "^18.2.0",

        "react-dom": "^18.2.0",

        "sass": "^1.3.0"

      },

      "peerDependenciesMeta": {

        "@opentelemetry/api": {

          "optional": true

        },

        "sass": {

          "optional": true

        }

      }

    },

    "node\_modules/nodemailer": {

      "version": "6.9.7",

      "resolved": "https://registry.npmjs.org/nodemailer/-/nodemailer-6.9.7.tgz",

      "integrity": "sha512-rUtR77ksqex/eZRLmQ21LKVH5nAAsVicAtAYudK7JgwenEDZ0UIQ1adUGqErz7sMkWYxWTTU1aeP2Jga6WQyJw==",

      "engines": {

        "node": ">=6.0.0"

      }

    },

    "node\_modules/picocolors": {

      "version": "1.0.0",

      "resolved": "https://registry.npmjs.org/picocolors/-/picocolors-1.0.0.tgz",

      "integrity": "sha512-1fygroTLlHu66zi26VoTDv8yRgm0Fccecssto+MhsZ0D/DGW2sm8E8AjW7NU5VVTRt5GxbeZ5qBuJr+HyLYkjQ=="

    },

    "node\_modules/postcss": {

      "version": "8.4.31",

      "resolved": "https://registry.npmjs.org/postcss/-/postcss-8.4.31.tgz",

      "integrity": "sha512-PS08Iboia9mts/2ygV3eLpY5ghnUcfLV/EXTOW1E2qYxJKGGBUtNjN76FYHnMs36RmARn41bC0AZmn+rR0OVpQ==",

      "funding": [

        {

          "type": "opencollective",

          "url": "https://opencollective.com/postcss/"

        },

        {

          "type": "tidelift",

          "url": "https://tidelift.com/funding/github/npm/postcss"

        },

        {

          "type": "github",

          "url": "https://github.com/sponsors/ai"

        }

      ],

      "dependencies": {

        "nanoid": "^3.3.6",

        "picocolors": "^1.0.0",

        "source-map-js": "^1.0.2"

      },

      "engines": {

        "node": "^10 || ^12 || >=14"

      }

    },

    "node\_modules/proxy-from-env": {

      "version": "1.1.0",

      "resolved": "https://registry.npmjs.org/proxy-from-env/-/proxy-from-env-1.1.0.tgz",

      "integrity": "sha512-D+zkORCbA9f1tdWRK0RaCR3GPv50cMxcrz4X8k5LTSUD1Dkw47mKJEZQNunItRTkWwgtaUSo1RVFRIG9ZXiFYg=="

    },

    "node\_modules/react": {

      "version": "18.2.0",

      "resolved": "https://registry.npmjs.org/react/-/react-18.2.0.tgz",

      "integrity": "sha512-/3IjMdb2L9QbBdWiW5e3P2/npwMBaU9mHCSCUzNln0ZCYbcfTsGbTJrU/kGemdH2IWmB2ioZ+zkxtmq6g09fGQ==",

      "dependencies": {

        "loose-envify": "^1.1.0"

      },

      "engines": {

        "node": ">=0.10.0"

      }

    },

    "node\_modules/react-dom": {

      "version": "18.2.0",

      "resolved": "https://registry.npmjs.org/react-dom/-/react-dom-18.2.0.tgz",

      "integrity": "sha512-6IMTriUmvsjHUjNtEDudZfuDQUoWXVxKHhlEGSk81n4YFS+r/Kl99wXiwlVXtPBtJenozv2P+hxDsw9eA7Xo6g==",

      "dependencies": {

        "loose-envify": "^1.1.0",

        "scheduler": "^0.23.0"

      },

      "peerDependencies": {

        "react": "^18.2.0"

      }

    },

    "node\_modules/scheduler": {

      "version": "0.23.0",

      "resolved": "https://registry.npmjs.org/scheduler/-/scheduler-0.23.0.tgz",

      "integrity": "sha512-CtuThmgHNg7zIZWAXi3AsyIzA3n4xx7aNyjwC2VJldO2LMVDhFK+63xGqq6CsJH4rTAt6/M+N4GhZiDYPx9eUw==",

      "dependencies": {

        "loose-envify": "^1.1.0"

      }

    },

    "node\_modules/source-map-js": {

      "version": "1.0.2",

      "resolved": "https://registry.npmjs.org/source-map-js/-/source-map-js-1.0.2.tgz",

      "integrity": "sha512-R0XvVJ9WusLiqTCEiGCmICCMplcCkIwwR11mOSD9CR5u+IXYdiseeEuXCVAjS54zqwkLcPNnmU4OeJ6tUrWhDw==",

      "engines": {

        "node": ">=0.10.0"

      }

    },

    "node\_modules/streamsearch": {

      "version": "1.1.0",

      "resolved": "https://registry.npmjs.org/streamsearch/-/streamsearch-1.1.0.tgz",

      "integrity": "sha512-Mcc5wHehp9aXz1ax6bZUyY5afg9u2rv5cqQI3mRrYkGC8rW2hM02jWuwjtL++LS5qinSyhj2QfLyNsuc+VsExg==",

      "engines": {

        "node": ">=10.0.0"

      }

    },

    "node\_modules/styled-jsx": {

      "version": "5.1.1",

      "resolved": "https://registry.npmjs.org/styled-jsx/-/styled-jsx-5.1.1.tgz",

      "integrity": "sha512-pW7uC1l4mBZ8ugbiZrcIsiIvVx1UmTfw7UkC3Um2tmfUq9Bhk8IiyEIPl6F8agHgjzku6j0xQEZbfA5uSgSaCw==",

      "dependencies": {

        "client-only": "0.0.1"

      },

      "engines": {

        "node": ">= 12.0.0"

      },

      "peerDependencies": {

        "react": ">= 16.8.0 || 17.x.x || ^18.0.0-0"

      },

      "peerDependenciesMeta": {

        "@babel/core": {

          "optional": true

        },

        "babel-plugin-macros": {

          "optional": true

        }

      }

    },

    "node\_modules/tslib": {

      "version": "2.6.2",

      "resolved": "https://registry.npmjs.org/tslib/-/tslib-2.6.2.tgz",

      "integrity": "sha512-AEYxH93jGFPn/a2iVAwW87VuUIkR1FVUKB77NwMF7nBTDkDrrT/Hpt/IrCJ0QXhW27jTBDcf5ZY7w6RiqTMw2Q=="

    },

    "node\_modules/watchpack": {

      "version": "2.4.0",

      "resolved": "https://registry.npmjs.org/watchpack/-/watchpack-2.4.0.tgz",

      "integrity": "sha512-Lcvm7MGST/4fup+ifyKi2hjyIAwcdI4HRgtvTpIUxBRhB+RFtUh8XtDOxUfctVCnhVi+QQj49i91OyvzkJl6cg==",

      "dependencies": {

        "glob-to-regexp": "^0.4.1",

        "graceful-fs": "^4.1.2"

      },

      "engines": {

        "node": ">=10.13.0"

      }

    }

  }

}

Next.config:

/\*\* @type {import('next').NextConfig} \*/

const nextConfig = {}

module.exports = nextConfig

Jsconfig:

{

  "compilerOptions": {

    "paths": {

      "@/\*": ["./src/\*"]

    }

  }

}