**TABLE OF CONTENTS**

Table of Contents

[Application outline 1](#_Toc153542817)

[1.1 Concept 1](#_Toc153542818)

[Architecture and data model diagrams 2](#_Toc153542819)

[2.1 High-level architecture diagram 2](#_Toc153542820)

[2.2 Data Model Diagram 3](#_Toc153542821)

[2.3 Security 3](#_Toc153542822)

[2.4 API 4](#_Toc153542823)

[2.5 User functionality 4](#_Toc153542824)

[2.6 Advanced Database Techniques 4](#_Toc153542825)

[3 Conclusion 5](#_Toc153542826)

GitHub Link: https://github.com/Mo1054/Mini-Project

# Application outline

## 1.1 Concept

Game Rating Web Application – RateThisGame

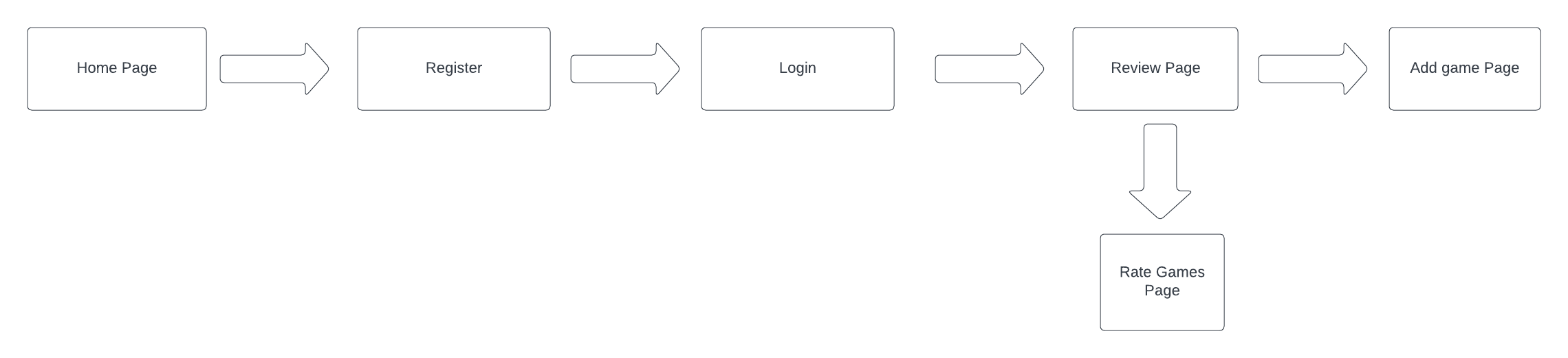
RateThisGame is a dynamic web application designed for users who wish to discover, rate, and review video games with minimal effort. Like rotten tomatoes and IMDB the web application will provide users with the ability to add their own opinion on a particular game. The users of the website will be able to make informed decisions based on the ratings of the games on the website. The rating system will work on a 5-star rating metric where users will be able to choose between 1 star and 5 stars. Users will be able to log in then add their own personal favourite games that will be rated by other users. Users will be required to log in via their email and passwords to leave a rating or a review.

**1.2 Project Scope**

**Key Features:**

* **Game Ratings and Reviews:** Rate and review games to help users make informed choices and uncover hidden gems.
* **Personalized Recommendations:** Discover new titles tailored to your preferences based on your ratings and interactions.
* **User-Generated Content:** Contribute to the gaming community by adding games and sharing your insights.
* **Community Engagement:** Connect with like-minded users, participate in discussions, and stay updated on gaming trends.

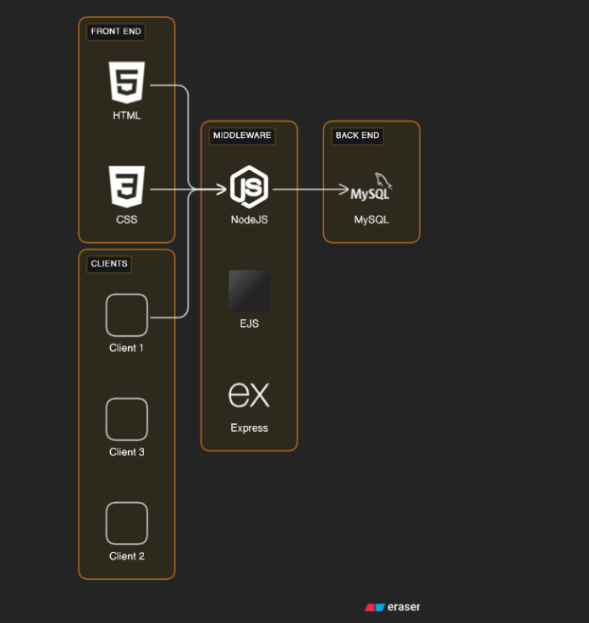
To help keep track of the project's development I’ve made a minimum viable product (MVP), I have created backlog to help prioritise what were the most key features needed for this web application to function as intended. Core functionality would look as follows:



# Architecture and data model diagrams

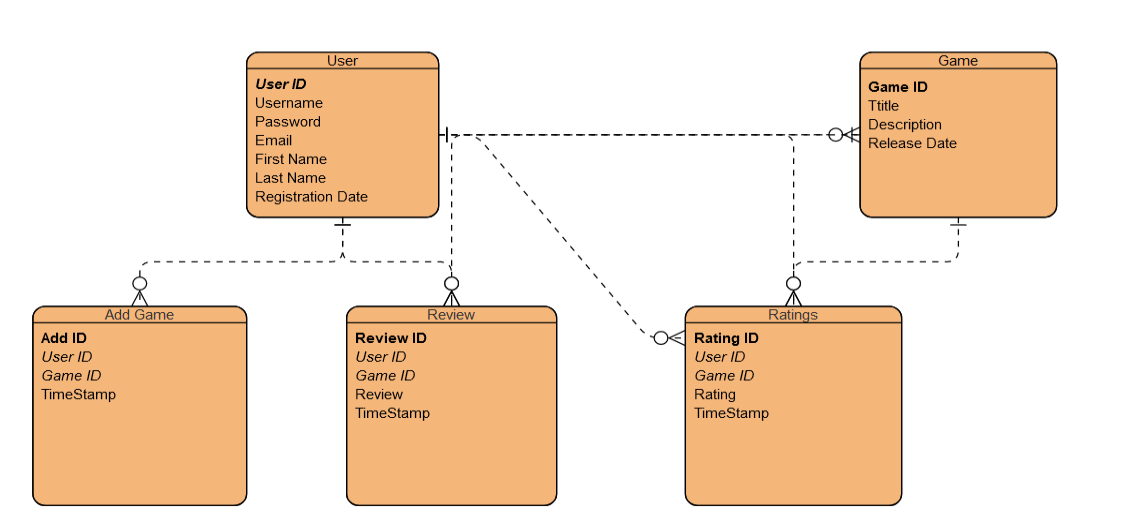
## 2.1 High-level architecture diagram

The diagram below shows all the components of the web application on each layer:



## 2.2 Data Model Diagram

The diagram below shows the entity relationship diagram and the column components for my web application.



# 2.3 Security

* **Password Hashing**: Used bcryptjs for hashing user passwords.
* **Session Management**: I used Express session for secure and efficient management of user sessions.
* **Input Validation and Sanitization**: I used input validation which is essential for preventing SQL injection and cross-site scripting (XSS) attacks.

# 2.4 API

* **Add Game**: POST /addGame - Adds a new game to the database. Requires game details in the request body.
* **View Game**: GET /viewGame - Retrieves details of a specific game.
* **Login and Registration**: POST endpoints for user authentication.
* **Home**: GET / - The home page route, displaying a list of games or user options.

# 2.5 User functionality

* **User Authentication**: Registration and login functionality, using bcryptjs for secure password handling.
* **Game Management**: Features for adding new games (addGame.js) and viewing game details (viewGame.js).
* **Navigation and Session Management**: Using Express sessions for maintaining user state across different pages like home (home.js), login (login.js), and registration (register.js).

# 2.6 Advanced Database Techniques

**Use of Data Types and Constraints**

* I used varying data types (int, varchar, date, timestamp) which is appropriate for the data stored, ensuring efficient space usage and query performance.
* I used primary key constraints on each table (id fields) ensure data uniqueness and integrity. The use of AUTO\_INCREMENT for these keys simplifies the process of adding new records.

**Timestamps for Record Tracking**

* Each table includes a createdAt timestamp. This is crucial for maintaining historical data and can be useful for data analysis purposes.

**Transaction Control for Data Integrity**

* I used of START TRANSACTION and COMMIT. This ensures that a group of SQL statements are executed in a way that the database remains consistent even in cases of system failures or interruptions.

**Alter Table and Queries**

* I used alter table commands and Queries. Queries are used to speed up item selection in databases.

# 3 Conclusion

In summary, RateThisGame is a dynamic web application designed to provide users with informative discussions and insightful game reviews. With its user-friendly 5-star rating system and key features like the ability to review games and even a dark mode option on all pages will create a platform where gamers can make informed decisions and engage in meaningful discussions about their favourite games.