**Testing Strategy and Scalability Plan**

This document is a write up to clarify the testing strategy and the scalability plan in order to ensure that Cinemate will be able to handle multiple users and works well under pressure

**Testing Strategy**

* **Unit testing:** Using Jest and React Testing Library will allow us to ensure the functionality of the components
* **Integration testing**: These tests will verify the interaction between the different parts of the application including API endpoints, UI, State management, etc.
* **Manual testing**: Using the manual testing will allow us to ensure the whole functionality of the web application by defining different scenarios and going through them

**Scalability plan**

* **Adding a backend:** Since this was a frontend take home test, Cinemate was built with no backend or a database, therefore if we want to up level Cinemate to handle multiple users it is a must to design a database and to add a backend to ensure that the users information’s are stored safely and to ensure the smooth flow of the information’s between the backend and the frontend
* **Horizontal Scaling:** Horizontal scaling will be achieved by adding more server instances or containers dynamically based on demand. Container orchestration platforms like Kubernetes will be considered for automated scaling and management.
* **Monitoring and Logging:** Setting up a logging and monitoring system is crucial to keep track of the health of the server and also to keep database usage and api calls
* **API Limiter:** As the number of the users grow, we need to setup an api limiter in order to not overpressure our server and protect it from breaking down