**Функционал frontend**

1. Интерфейс поиска фильмов

- Поле поиска: вместо универсального текстового поля для ввода ключевых слов создадим набор выпадающих меню (селектов) и чекбоксов для выбора конкретных параметров поиска.

- Жанр: выпадающее меню с возможностью множественного выбора (чекбоксы или многоуровневые селекты) для фильтрации по жанрам.

- Год выпуска: числовой селект или диапазон, чтобы пользователь мог выбрать конкретный год или период (например, «2010–2020»).

- Страна: выпадающий список с возможностью поиска по названию страны.

- Рейтинг: можно добавить ползунок или селектор с диапазоном рейтинга для выбора минимального и максимального рейтинга.

- Язык: выпадающее меню для выбора языка фильма.

- Фильтры и Категории:

- Вместе с полями для параметров поиска можно добавить кнопки категорий, такие как «Популярные», «Новинки», «Лучшие по рейтингу». Эти кнопки можно разместить отдельно, чтобы сразу показать популярные подборки.

- Добавить кнопку «Сбросить фильтры», чтобы быстро сбросить все выбранные параметры.

- Кнопка поиска: после выбора параметров пользователю нужно нажать кнопку «Найти» для отображения результатов, соответствующих выбранным фильтрам.

2. Реализация интерфейса поиска

Для фронтенд-разработки таких фильтров можно использовать библиотеки и инструменты, которые упростят работу:

- React Select: для выпадающих списков с возможностью выбора нескольких значений (например, жанры или страны).

- Range Slider (например, Material-UI Slider или react-slider) для выбора диапазона (например, рейтинга или года выпуска).

- Formik или React Hook Form для управления состоянием формы и сбора всех значений фильтров.

- API для поиска: при нажатии на кнопку «Найти» отправляется запрос на сервер, где параметры (жанр, страна и т. д.) передаются как query-параметры API-запроса.

**Принцип работы**

1. Выбор фильтров: пользователь выбирает необходимые параметры с помощью выпадающих меню и слайдеров.

2. Формирование запроса: при нажатии кнопки «Найти» значения всех полей собираются и передаются на сервер для фильтрации результатов.

3. Отображение результатов: сервер возвращает фильмы, соответствующие критериям, которые затем выводятся на странице с результатами.

4. Страница с информацией о фильме

- Основная информация: реализация компонента карточки с основной информацией (например, синопсис, рейтинг, постеры) и прокрутка для раздела с трейлером.

- Отзывы и рейтинги: интеграция компонентов формы с рейтингами и комментариями (например, React Rating Stars).

- Кнопки действий: поддержка сохранения в избранное и просмотра трейлера, можно использовать сервисы Firebase или localStorage для хранения избранного пользователя.

5. Адаптивный дизайн

- Разработка UI с применением CSS Flexbox/Grid для отзывчивой верстки.

- Для адаптивности и отзывчивости можно использовать CSS media queries, а также такие CSS-фреймворки, как Tailwind CSS или Bootstrap.

- Для мобильных устройств стоит добавить поддержку свайпов и жестов (например, react-swipeable для React).

6. Регистрация и авторизация пользователей

- Формы регистрации/входа: Firebase Authentication или OAuth для безопасного входа.

- Личный кабинет: возможность редактирования профиля, добавление в избранное и просмотр истории — через управление состоянием в контексте пользователя.

- Уведомления: push-уведомления можно реализовать с помощью Firebase Cloud Messaging или аналогичных сервисов.

7. Реализация отзывчивости и интерактивности

- Анимации и переходы: библиотеки для анимаций, такие как Framer Motion (React), помогут создать плавные переходы