**Критерии приема**

Используй этот [шаблон React-проекта](https://github.com/goitacademy/react-homework-template#readme) как стартовую точку своего приложения.

* Создан репозиторий goit-react-hw-05-movies
* При сдаче домашней работы есть ссылки: на исходные файлы и рабочие страницы каждого проекта на GitHub Pages
* В состоянии компонентов хранится минимально необходимый набор данных, остальное вычисляется
* При запуске кода задания, в консоли нет ошибок и предупреждений
* Для каждого компонента есть отдельная папка с файлом React-компонента и файлом стилей
* Все что компонент ожидает в виде пропов, передается ему при вызове
* Имена компонентов понятные, описательные
* JS-код чистый и понятный, используется Prettier
* Стилизация выполнена CSS-модулями или Styled Components.

**Кинопоиск**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-05/#%D0%BA%D0%B8%D0%BD%D0%BE%D0%BF%D0%BE%D0%B8%D1%81%D0%BA)

Создай базовую маршрутизацию для приложения поиска и хранения фильмов. Превью рабочего приложения [смотри по ссылке](https://drive.google.com/file/d/1vR0hi3n1236Q5Bg4-se-8JVKD9UKSfId/view?usp=sharing).

**API themoviedb.org**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-05/#api-themoviedborg)

Для бекенда используй [themoviedb.org API](https://www.themoviedb.org/). Необходимо зарегистриваться (можно ввести произвольные данные) и получить API-ключ. В этой работе будут использоваться следующие ендпоинты.

* [/trending/get-trending](https://developers.themoviedb.org/3/trending/get-trending) список самых популярных фильмов на сегодня для создания коллекции на главной странице.
* [/search/search-movies](https://developers.themoviedb.org/3/search/search-movies) поиск кинофильма по ключевому слову на странице фильмов.
* [/movies/get-movie-details](https://developers.themoviedb.org/3/movies/get-movie-details) запрос полной информации о фильме для страницы кинофильма.
* [/movies/get-movie-credits](https://developers.themoviedb.org/3/movies/get-movie-credits) запрос информации о актёрском составе для страницы кинофильма.
* [/movies/get-movie-reviews](https://developers.themoviedb.org/3/movies/get-movie-reviews) запрос обзоров для страницы кинофильма.

[Ссылка на документацию](https://developers.themoviedb.org/3/getting-started/introduction)

**Маршруты**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-05/#%D0%BC%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D1%8B)

В приложении должны быть следующие маршруты. Если пользователь зашел по несуществующему маршруту, его необходимо перенаправлять на домашнюю страницу.

* '/' - компонент Home, домашняя страница со списком популярных кинофильмов.
* '/movies' - компонент Movies, страница поиска фильмов по ключевому слову.
* '/movies/:movieId' - компонент MovieDetails, страница с детальной информацией о кинофильме.
* /movies/:movieId/cast - компонент Cast, информация о актерском составе. Рендерится на странице MovieDetails.
* /movies/:movieId/reviews - компонент Reviews, информация об обзорах. Рендерится на странице MovieDetails.

**Code Splitting (разделение кода)**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-05/#code-splitting-%D1%80%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5-%D0%BA%D0%BE%D0%B4%D0%B0)

Добавь асинхронную загрузку JS-кода для маршрутов приложения используя React.lazy() и <Suspense>.