# Frontend Development Notes

## React & TypeScript

The frontend of our Movie Picture application is written in TypeScript and uses the React framework. This means that the codebase adheres to strict type-checking and a component-based structure.

## eslint

This project uses eslint for code quality. It's important that all code adheres to the rules outlined in our `.eslintrc` file. The linter will automatically check your code for style issues, potential bugs, and enforce certain design principles.

## React Testing Library

Our application uses the React Testing Library for unit testing. This testing library is focused on the user's perspective. The tests are designed to resemble how users interact with your app.

## GitHub Actions

GitHub Actions are used to automate our software development workflows. GitHub Actions will be responsible for running our linter, tests, and building the app whenever there is a `pull_request` against the `main` branch. It will also handle the deployment of our app whenever there is a `push` to the `main` branch.

## Docker

We're using Docker to containerize our frontend application.

## Kubernetes

Deployment of our app to the existing Kubernetes cluster will be automated by our GitHub Actions workflows.

## AWS & Terraform

We're using AWS to host our Kubernetes cluster and Terraform to manage our infrastructure as code. You'll need to create AWS infrastructure using the Terraform scripts provided. Follow the instructions in the exercise carefully and ensure you have the necessary permissions to perform these actions.

As you work on this project, remember to focus on understanding each part of the pipeline. Make sure that all your workflows are correctly configured and that they trigger as expected. Keep the best practices in mind as you work and ensure that your code is clean and well-tested.