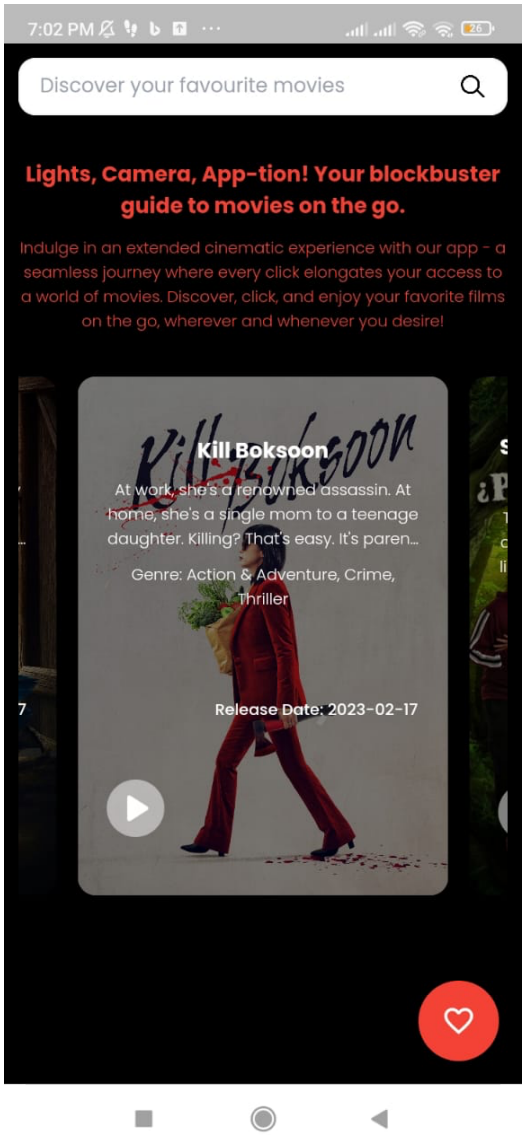
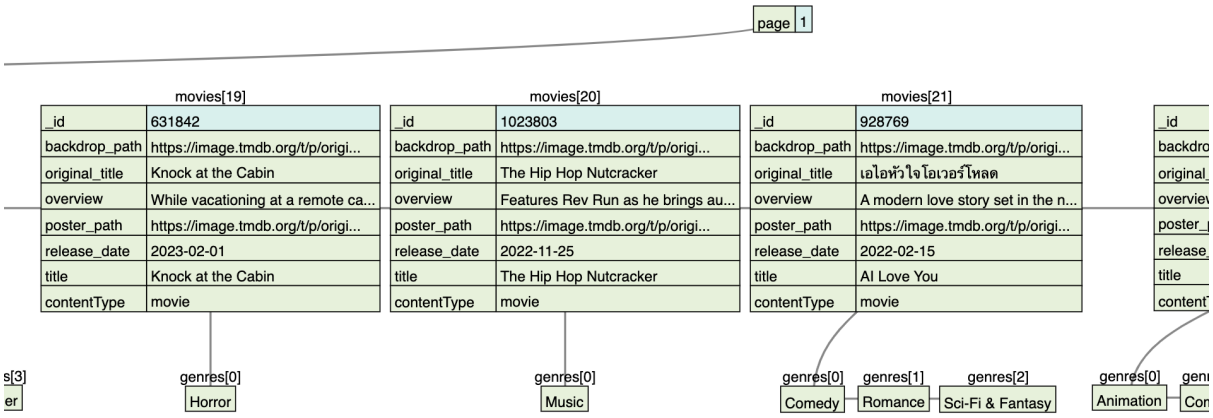


On this initial screen I pre fetch data from following api via use of dio package and getx controllers for managing the data, search isn't functional.

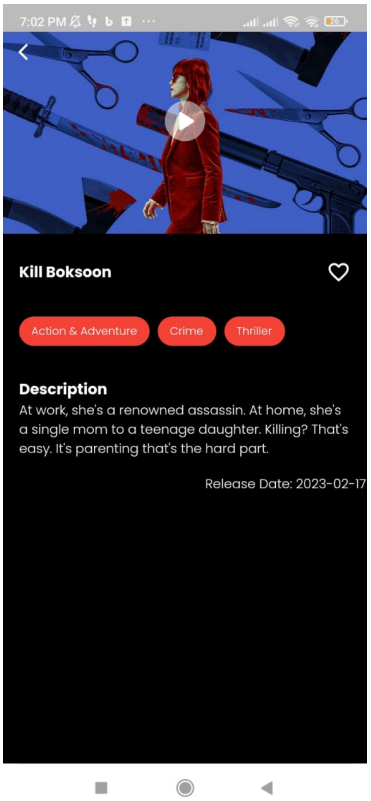
Base Url	**https://movies-api14.p.rapidapi.com**
End point	**/movies**
Request Type	**GET**
Headers	**{'X-RapidAPI-Host': 'movies-api14.p.rapidapi.com', 'X-RapidAPI-Key': 'baf41e5371mshc97f4d28e95a407p112358jsnd26e90d153a1', }**

Example Response:

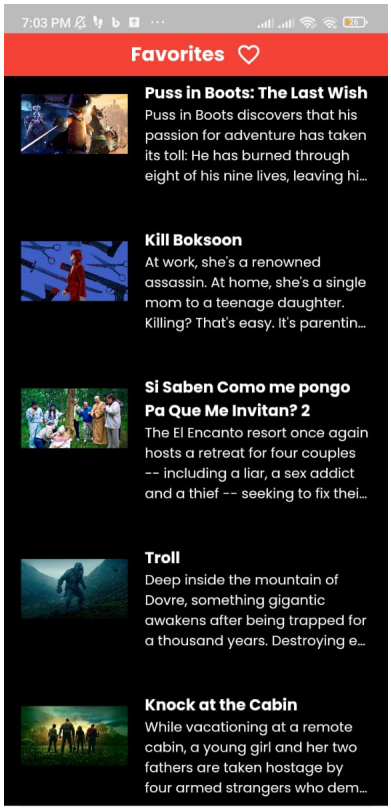


Afterwards the response is parsed using movies model which look a little something like

```
class MoviesModel {
    List<Movie>? movies;
    int? page;

    MoviesModel({
        this.movies,
        this.page,
    });

    MoviesModel copyWith({
        List<Movie>? movies,
        int? page,
    }) =>
        MoviesModel(
            movies: movies ?? this.movies,
            page: page ?? this.page,
        );
}

class Movie {
    int? id;
    String? backdropPath;
    List<String>? genres;
    String? originalTitle;
    String? overview;
    String? posterPath;
    DateTime? releaseDate;
    String? title;
    ContentType? contentType;

    Movie({
        this.id,
        this.backdropPath,
        this.genres,
        this.originalTitle,
        this.overview,
        this.posterPath,
        this.releaseDate,
        this.title,
        this.contentType,
    });

    Movie copyWith({
        int? id,
        String? backdropPath,
        List<String>? genres,
        String? originalTitle,
        String? overview,
        String? posterPath,
        DateTime? releaseDate,
        String? title,
        ContentType? contentType,
    }) =>
        Movie(
            id: id ?? this.id,
            backdropPath: backdropPath ?? this.backdropPath,
            genres: genres ?? this.genres,
            originalTitle: originalTitle ?? this.originalTitle,
            overview: overview ?? this.overview,
            posterPath: posterPath ?? this.posterPath,
            releaseDate: releaseDate ?? this.releaseDate,
            title: title ?? this.title,
            contentType: contentType ?? this.contentType,
        );
}

enum ContentType {
    MOVIE
}
```



Next on this screen no new api was used instead the data at that specific index was passed as an argument to the next screen using getx.



Lastly, for this screen a getx controller was made which used shared preferences to locally store model of favorited movie and this controller was injected in main function after widget binding to fetch favorites while starting the app. This controller was further comprising of function to save the model load the model check if a movie is favorited and lastly remove from favorites.