Hobby project – Movie watch list

Harry Emmett

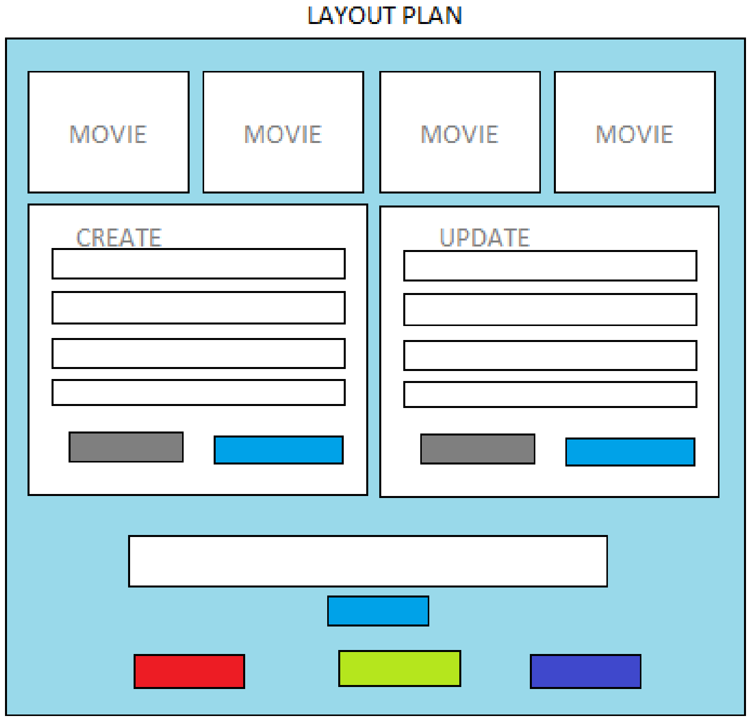
Introduction and technologies used

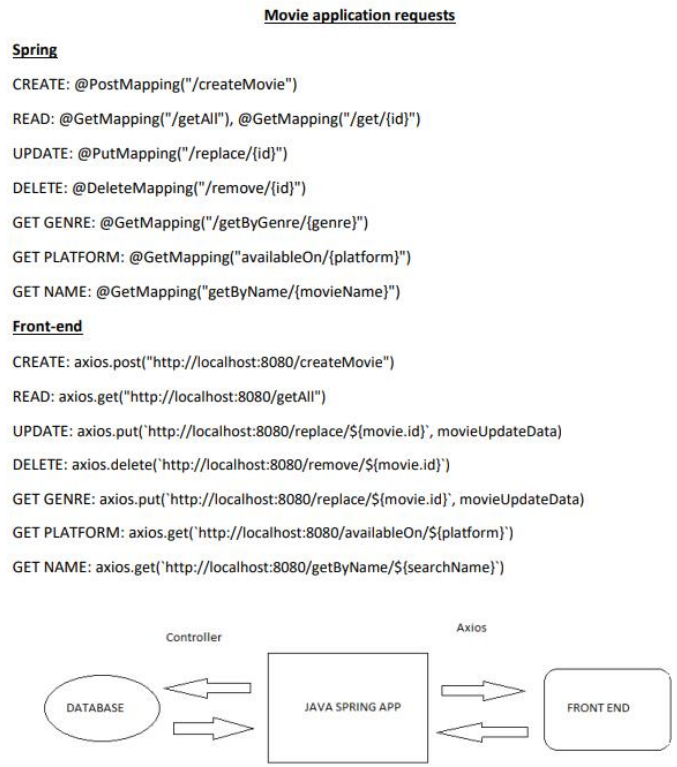
- Backend Language: Java + Spring framework
- Frontend Languages: JavaScript, HTML, CSS with Bootstrap framework
- Version control: Git / GitHub
- Tests: Mock MVC controller integration test
- Database: mySQL + H2
- Project management: Jira
- Other: Postman

How I approached the project

- Planning – Jira board, Mock webpage, MoSCow, Application design flow diagram.
- Risk assessment.
- Version control.
- Backend build.
- Frontend build.
- Time management.

Planning and preparation





Movie application requests

Spring

CREATE: @PostMapping("/createMovie")

READ: @GetMapping("/getAll"), @GetMapping("/get/{id}")

UPDATE: @PutMapping("/replace/{id}")

DELETE: @DeleteMapping("/remove/{id}")

GET GENRE: @GetMapping("/getByGenre/{genre}")

GET PLATFORM: @GetMapping("availableOn/{platform}")

GET NAME: @GetMapping("getByName/{movieName}")

Front-end

CREATE: axios.post("http://localhost:8080/createMovie")

READ: axios.get("http://localhost:8080/getAll")

UPDATE: axios.put(`http://localhost:8080/replace/${movie.id}`, movieUpdateData)

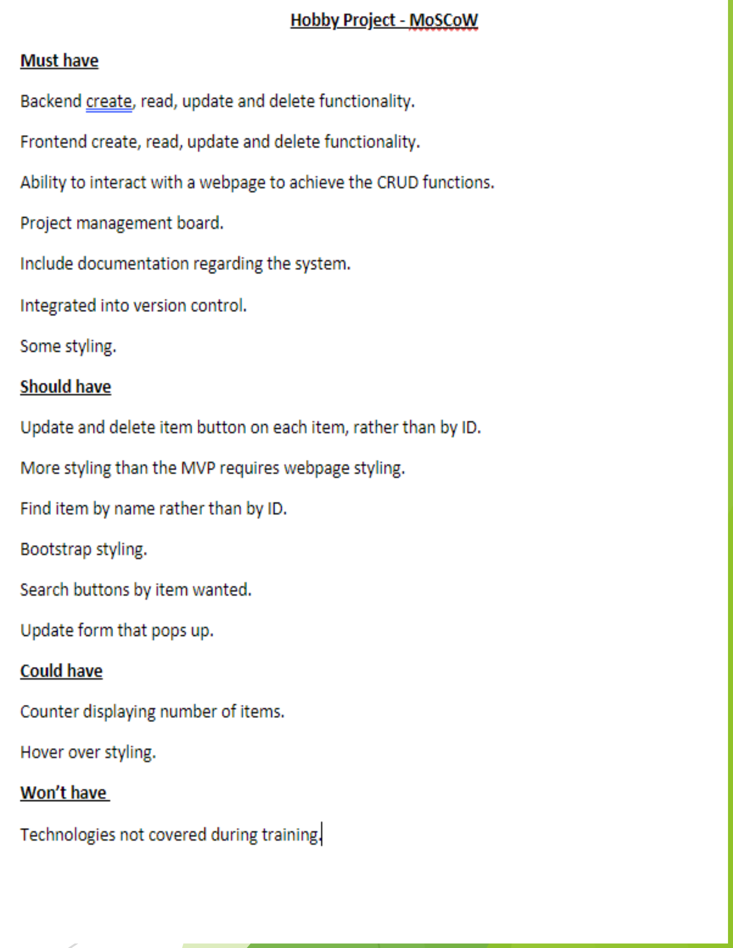
DELETE: axios.delete(`http://localhost:8080/remove/${movie.id}`)

GET GENRE: axios.put(`http://localhost:8080/replace/${movie.id}`, movieUpdateData)

GET PLATFORM: axios.get(`http://localhost:8080/availableOn/${platform}`)

GET NAME: axios.get(`http://localhost:8080/getByName/${searchName}`)

Controller
Axios
DATABASE
JAVA SPRING APP
FRONT END



Hobby Project - MoSCoW

Must have

Backend create, read, update and delete functionality.

Frontend create, read, update and delete functionality.

Ability to interact with a webpage to achieve the CRUD functions.

Project management board.

Include documentation regarding the system.

Integrated into version control.

Some styling.

Should have

Update and delete item button on each item, rather than by ID.

More styling than the MVP requires webpage styling.

Find item by name rather than by ID.

Bootstrap styling.

Search buttons by item wanted.

Update form that pops up.

Could have

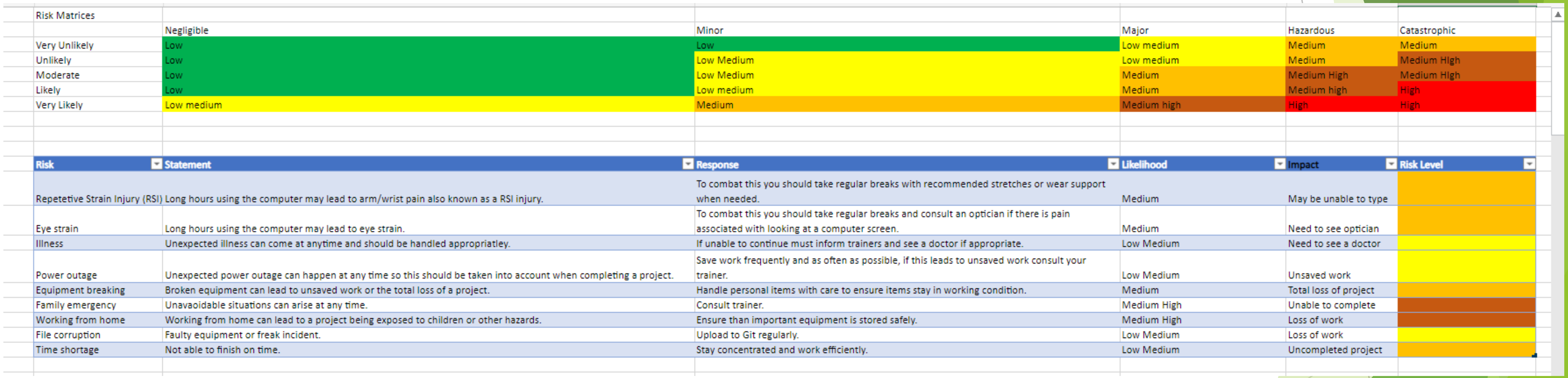
Counter displaying number of items.

Hover over styling.

Won't have

Technologies not covered during training.

Risk assessment



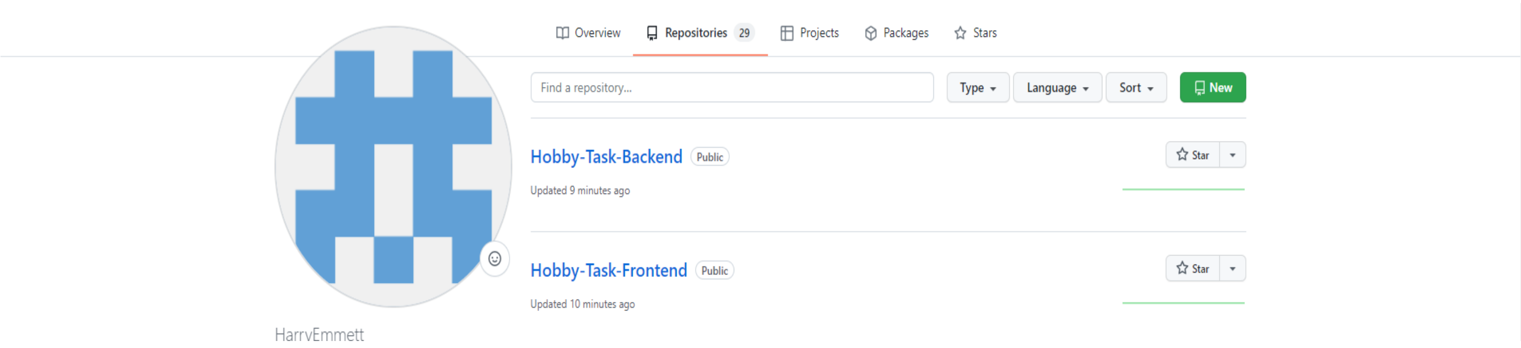
Risk Matrices

	Negligible	Minor	Major	Hazardous	Catastrophic
Very Unlikely	Low	Low	Low medium	Medium	Medium
Unlikely	Low	Low Medium	Low medium	Medium	Medium HIgh
Moderate	Low	Low Medium	Medium	Medium High	Medium HIgh
Likely	Low	Low medium	Medium	Medium high	High
Very Likely	Low medium	Medium	Medium high	High	High

Risk	Statement	Response	Likelihood	Impact	Risk Level
Repetetive Strain Injury (RSI)	Long hours using the computer may lead to arm/wrist pain also known as a RSI injury.	To combat this you should take regular breaks with recommended stretches or wear support when needed.	Medium	May be unable to type	🟧
Eye strain	Long hours using the computer may lead to eye strain.	To combat this you should take regular breaks and consult an optician if there is pain associated with looking at a computer screen.	Medium	Need to see optician	🟧
Illness	Unexpected illness can come at anytime and should be handled appropriatley.	If unable to continue must inform trainers and see a doctor if appropriate.	Low Medium	Need to see a doctor	🟨
Power outage	Unexpected power outage can happen at any time so this should be taken into account when completing a project.	Save work frequently and as often as possible, if this leads to unsaved work consult your trainer.	Low Medium	Unsaved work	🟨
Equipment breaking	Broken equipment can lead to unsaved work or the total loss of a project.	Handle personal items with care to ensure items stay in working condition.	Medium	Total loss of project	🟧
Family emergency	Unavaoidable situations can arise at any time.	Consult trainer.	Medium High	Unable to complete	🟫
Working from home	Working from home can lead to a project being exposed to children or other hazards.	Ensure than important equipment is stored safely.	Medium High	Loss of work	🟫
File corruption	Faulty equipment or freak incident.	Upload to Git regularly.	Low Medium	Loss of work	🟨
Time shortage	Not able to finish on time.	Stay concentrated and work efficiently.	Low Medium	Uncompleted project	🟧

Version control

Two repos used



HarrvEmmett

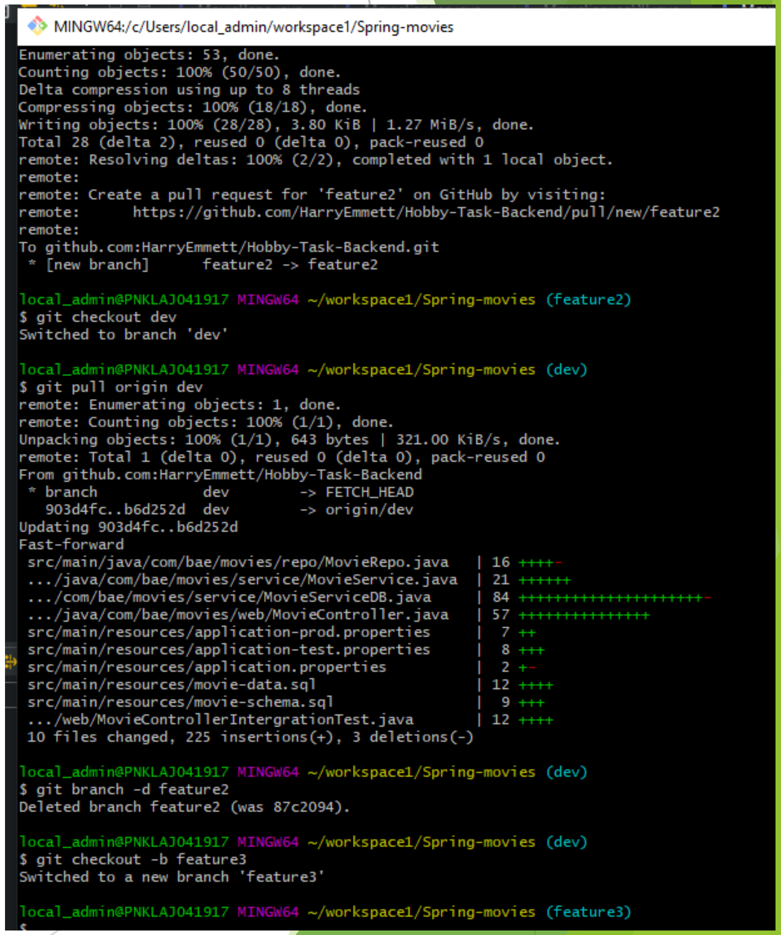
Hobby-Task-Backend Public

Updated 9 minutes ago

Hobby-Task-Frontend Public

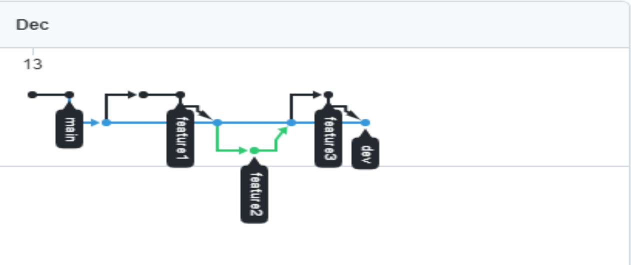
Updated 10 minutes ago

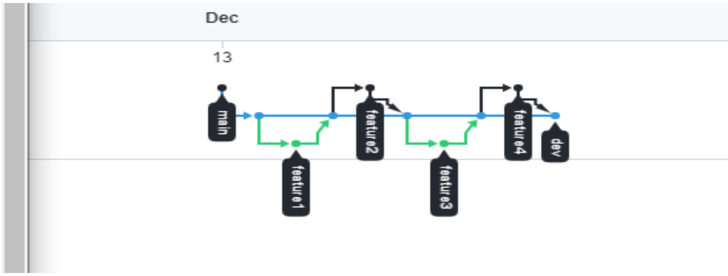
FMB and deleting branches

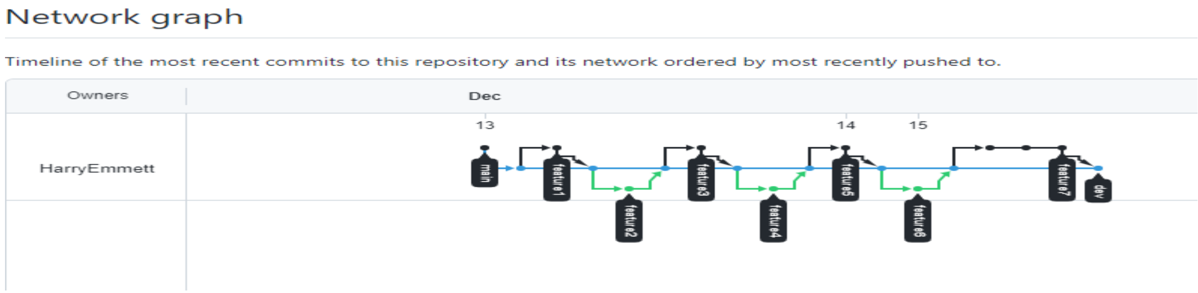


```
Enumerating objects: 53, done.
Counting objects: 100% (50/50), done.
Delta compression using up to 8 threads
Compressing objects: 100% (18/18), done.
Writing objects: 100% (28/28), 3.80 KiB | 1.27 MiB/s, done.
Total 28 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
remote:
remote: Create a pull request for 'feature2' on GitHub by visiting:
remote:      https://github.com/HarryEmmett/Hobby-Task-Backend/pull/new/feature2
remote:
To github.com:HarryEmmett/Hobby-Task-Backend.git
 * [new branch]      feature2 -> feature2

local_admin@PNKLAJ041917 MINGW64 ~/workspace1/Spring-movies (feature2)
$ git checkout dev
Switched to branch 'dev'

local_admin@PNKLAJ041917 MINGW64 ~/workspace1/Spring-movies (dev)
$ git pull origin dev
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
Unpacking objects: 100% (1/1), 643 bytes | 321.00 KiB/s, done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
From github.com:HarryEmmett/Hobby-Task-Backend
 * branch            dev        -> FETCH_HEAD
   903d4fc..b6d252d  dev        -> origin/dev
Updating 903d4fc..b6d252d
Fast-forward
 src/main/java/com/bae/movies/repo/MovieRepo.java   | 16 ++++-
 .../java/com/bae/movies/service/MovieService.java  | 21 ++++++
 .../com/bae/movies/service/MovieServiceDB.java     | 84 +++++++++++++++++++++-
 .../java/com/bae/movies/web/MovieController.java   | 57 +++++++++++++++
 src/main/resources/application-prod.properties     |  7 ++
 src/main/resources/application-test.properties     |  8 +++
 src/main/resources/application.properties          |  2 +-
 src/main/resources/movie-data.sql                  | 12 ++++
 src/main/resources/movie-schema.sql                |  9 +++
 .../web/MovieControllerIntergrationTest.java       | 12 ++++
 10 files changed, 225 insertions(+), 3 deletions(-)

local_admin@PNKLAJ041917 MINGW64 ~/workspace1/Spring-movies (dev)
$ git branch -d feature2
Deleted branch feature2 (was 87c2094).

local_admin@PNKLAJ041917 MINGW64 ~/workspace1/Spring-movies (dev)
$ git checkout -b feature3
Switched to a new branch 'feature3'

local_admin@PNKLAJ041917 MINGW64 ~/workspace1/Spring-movies (feature3)
$
```

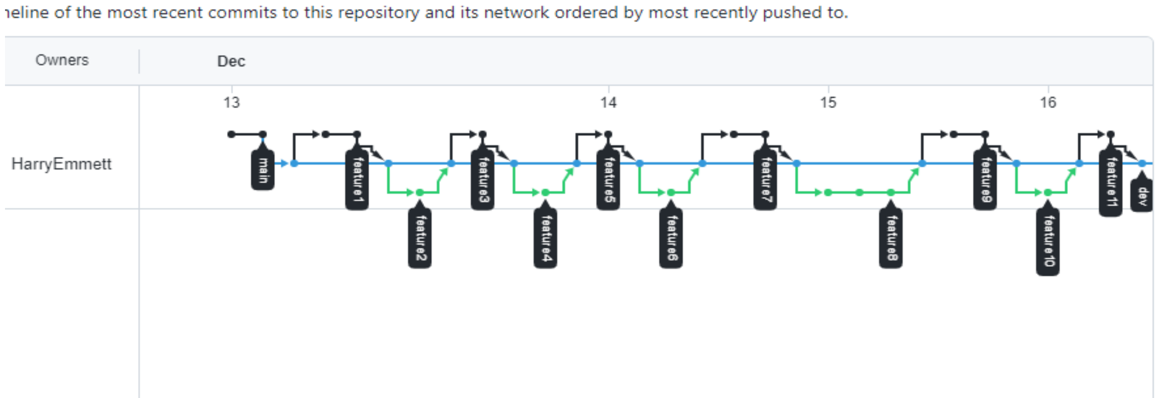
Feature branch model



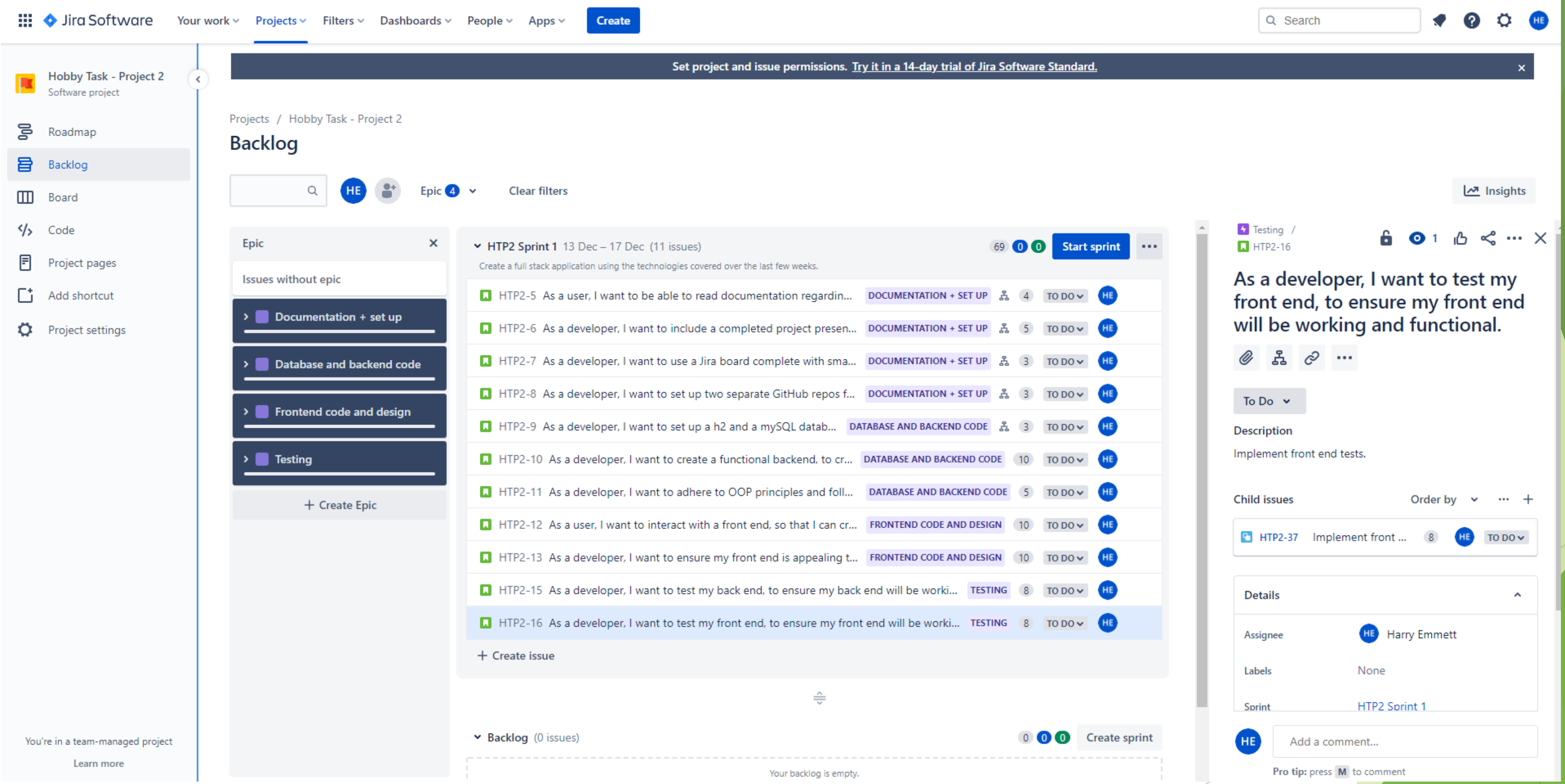



Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.


ıeline of the most recent commits to this repository and its network ordered by most recently pushed to.

Jira



Set project and issue permissions. Try it in a 14-day trial of Jira Software Standard.

Hobby Task - Project 2
Software project

You're in a team-managed project
Learn more

Projects / Hobby Task - Project 2

Backlog

Epic

Issues without epic

- Documentation + set up
- Database and backend code
- Frontend code and design
- Testing

HTP2 Sprint 1 13 Dec – 17 Dec (11 issues) 69 0 0

Create a full stack application using the technologies covered over the last few weeks.

Key	Summary	Epic	Points	Status	Assignee
HTP2-5	As a user, I want to be able to read documentation regardin...	DOCUMENTATION + SET UP	4	TO DO	HE
HTP2-6	As a developer, I want to include a completed project presen...	DOCUMENTATION + SET UP	5	TO DO	HE
HTP2-7	As a developer, I want to use a Jira board complete with sma...	DOCUMENTATION + SET UP	3	TO DO	HE
HTP2-8	As a developer, I want to set up two separate GitHub repos f...	DOCUMENTATION + SET UP	3	TO DO	HE
HTP2-9	As a developer, I want to set up a h2 and a mySQL datab...	DATABASE AND BACKEND CODE	3	TO DO	HE
HTP2-10	As a developer, I want to create a functional backend, to cr...	DATABASE AND BACKEND CODE	10	TO DO	HE
HTP2-11	As a developer, I want to adhere to OOP principles and foll...	DATABASE AND BACKEND CODE	5	TO DO	HE
HTP2-12	As a user, I want to interact with a front end, so that I can cr...	FRONTEND CODE AND DESIGN	10	TO DO	HE
HTP2-13	As a developer, I want to ensure my front end is appealing t...	FRONTEND CODE AND DESIGN	10	TO DO	HE
HTP2-15	As a developer, I want to test my back end, to ensure my back end will be worki...	TESTING	8	TO DO	HE
HTP2-16	As a developer, I want to test my front end, to ensure my front end will be worki...	TESTING	8	TO DO	HE

Backlog (0 issues) 0 0 0

Your backlog is empty.

Testing / HTP2-16

As a developer, I want to test my front end, to ensure my front end will be working and functional.

Description

Implement front end tests.

Child issues

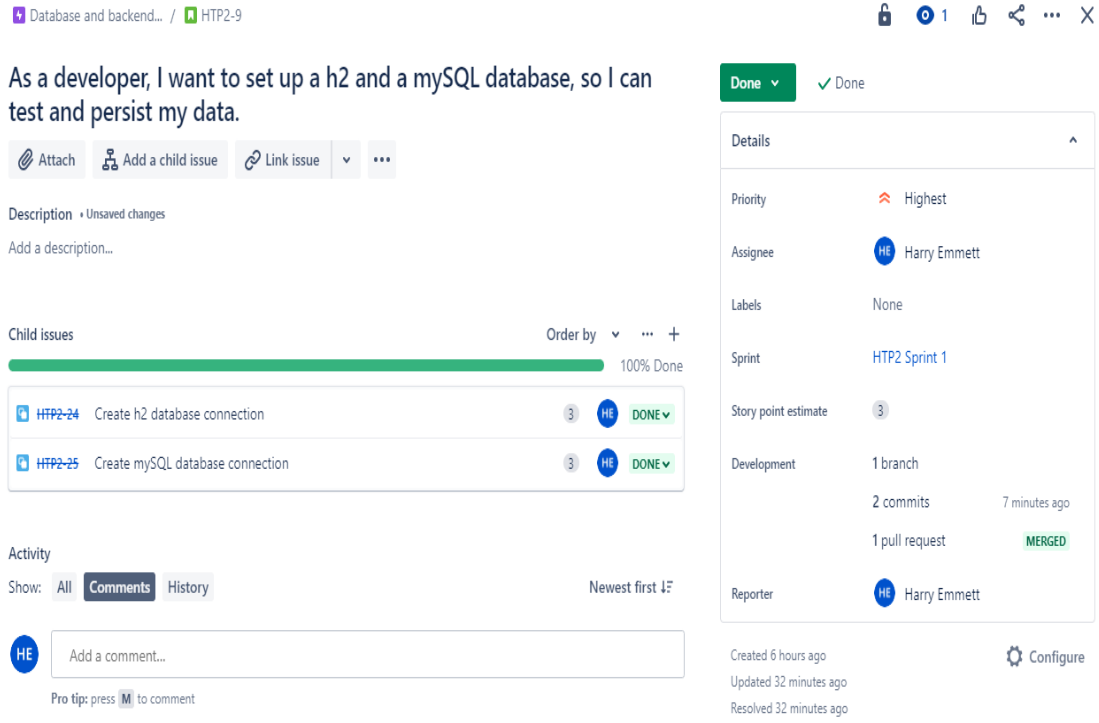
HTP2-37	Implement front ...	8	HE	TO DO

Details

Assignee	Harry Emmett
Labels	None
Sprint	HTP2 Sprint 1

Pro tip: press M to comment

Jira continued – smart commits



Database and backend... / HTP2-9

As a developer, I want to set up a h2 and a mySQL database, so I can test and persist my data.

Description • Unsaved changes

Add a description...

Child issues

100% Done

~~HTP2-24~~	Create h2 database connection	3	HE	DONE
~~HTP2-25~~	Create mySQL database connection	3	HE	DONE

Activity

Show: All Comments History — Newest first

Add a comment...

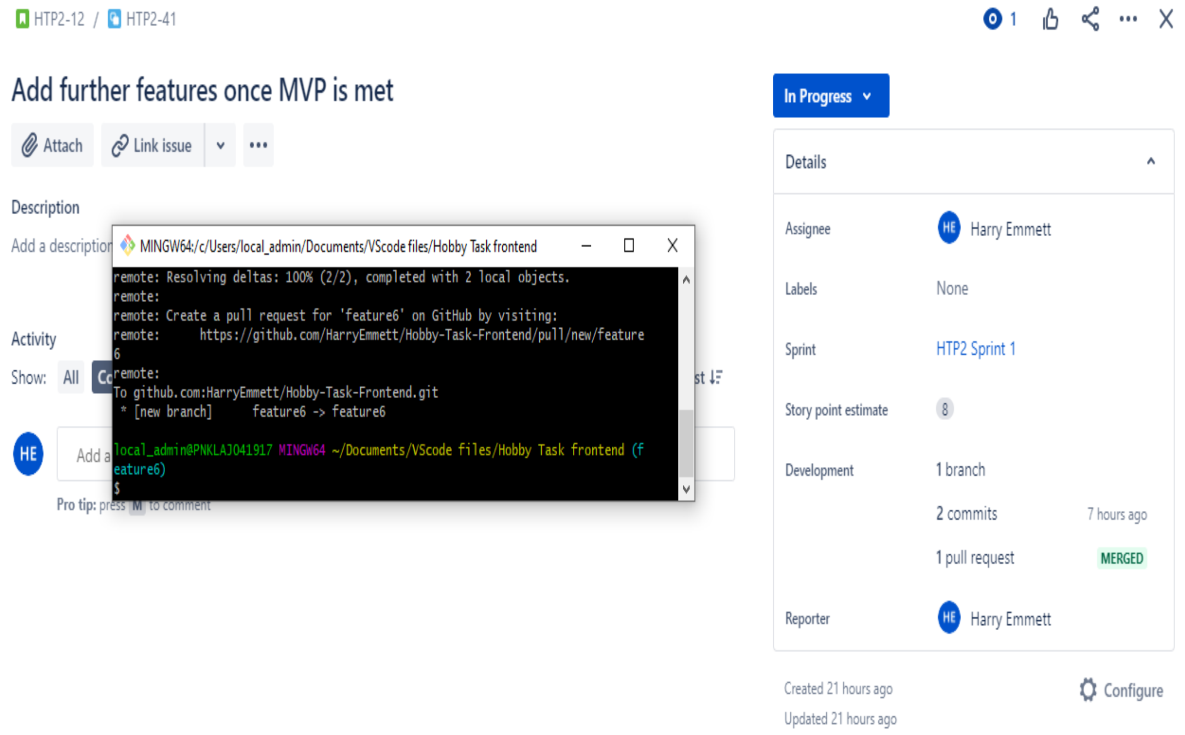
Pro tip: press M to comment

Done ✓ Done

Details

Priority	Highest	
Assignee	Harry Emmett	
Labels	None	
Sprint	HTP2 Sprint 1	
Story point estimate	3	
Development	1 branch	
	2 commits	7 minutes ago
	1 pull request	MERGED
Reporter	Harry Emmett	

Created 6 hours ago
Updated 32 minutes ago
Resolved 32 minutes ago



HTP2-12 / HTP2-41

Add further features once MVP is met

Description

Add a description...

Activity

Show: All

Pro tip: press M to comment

MINGW64:/c/Users/local_admin/Documents/VScode files/Hobby Task frontend

```
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'feature6' on GitHub by visiting:
remote:      https://github.com/HarryEmmett/Hobby-Task-Frontend/pull/new/feature
6
remote:
To github.com:HarryEmmett/Hobby-Task-Frontend.git
 * [new branch]      feature6 -> feature6

local_admin@PNKLAJ041917 MINGW64 ~/Documents/VScode files/Hobby Task frontend (f
eature6)
$
```

In Progress

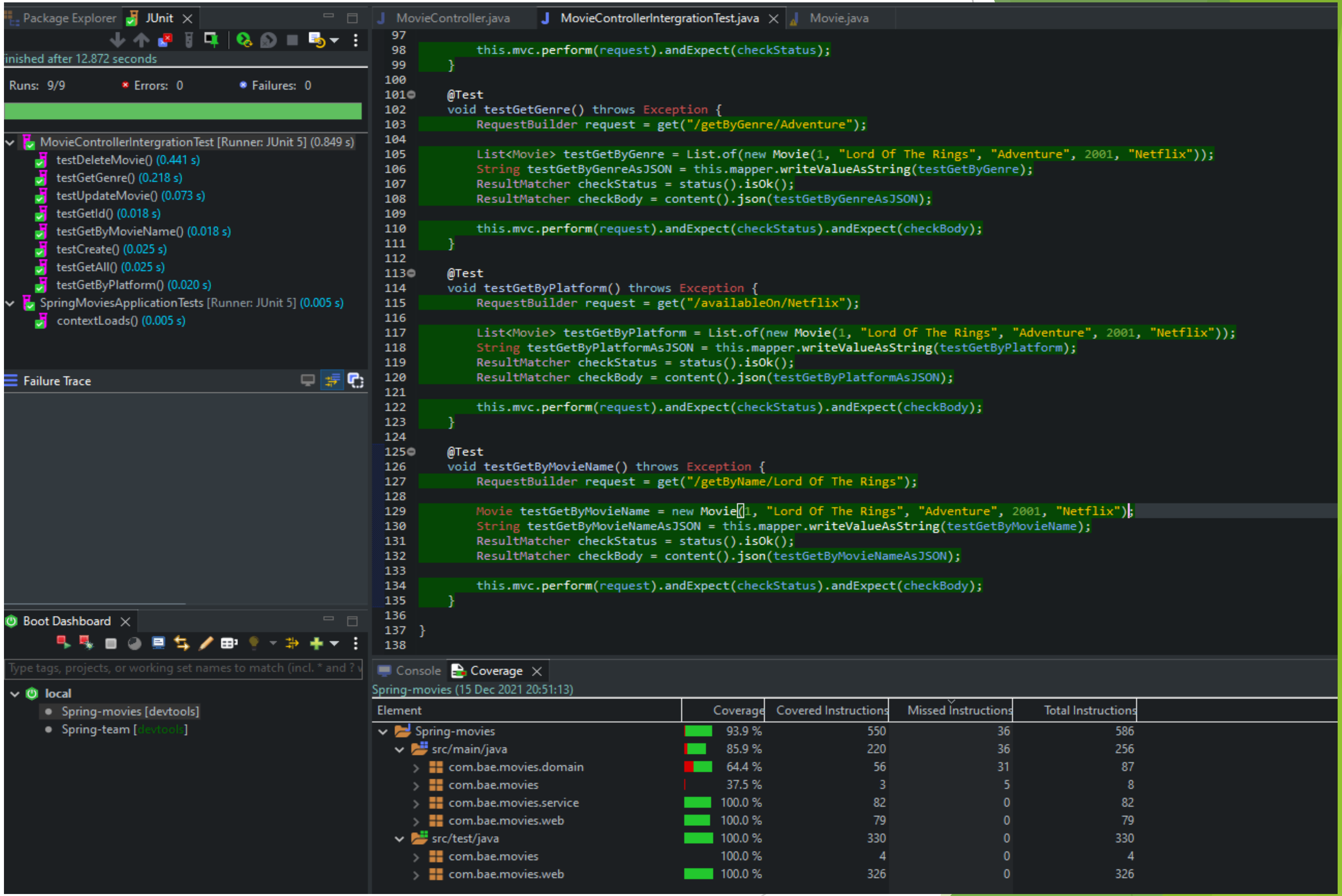
Details

Assignee	Harry Emmett	
Labels	None	
Sprint	HTP2 Sprint 1	
Story point estimate	8	
Development	1 branch	
	2 commits	7 hours ago
	1 pull request	MERGED
Reporter	Harry Emmett	

Created 21 hours ago
Updated 21 hours ago

Testing

- MVC controller tests



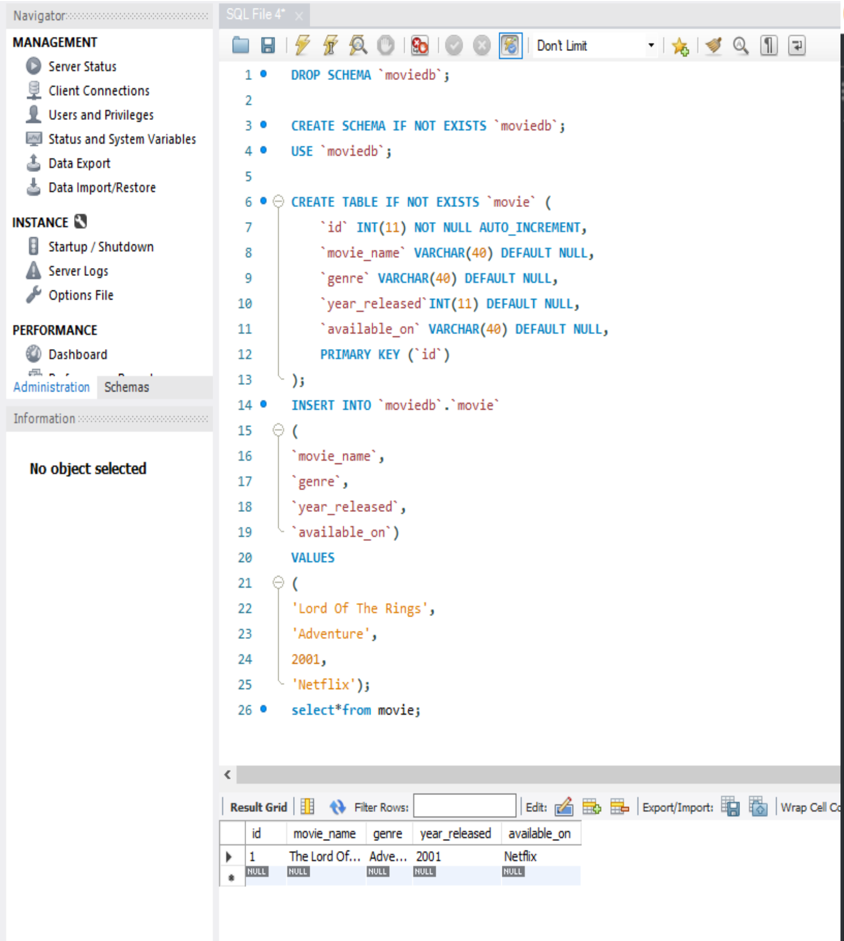
```
inished after 12.872 seconds
Runs:  9/9        Errors:  0        Failures:  0

MovieControllerIntergrationTest [Runner: JUnit 5] (0.849 s)
    testDeleteMovie() (0.441 s)
    testGetGenre() (0.218 s)
    testUpdateMovie() (0.073 s)
    testGetId() (0.018 s)
    testGetByMovieName() (0.018 s)
    testCreate() (0.025 s)
    testGetAll() (0.025 s)
    testGetByPlatform() (0.020 s)
SpringMoviesApplicationTests [Runner: JUnit 5] (0.005 s)
    contextLoads() (0.005 s)
```

```java

        this.mvc.perform(request).andExpect(checkStatus);
    }

    @Test
    void testGetGenre() throws Exception {
        RequestBuilder request = get("/getByGenre/Adventure");

        List<Movie> testGetByGenre = List.of(new Movie(1, "Lord Of The Rings", "Adventure", 2001, "Netflix"));
        String testGetByGenreAsJSON = this.mapper.writeValueAsString(testGetByGenre);
        ResultMatcher checkStatus = status().isOk();
        ResultMatcher checkBody = content().json(testGetByGenreAsJSON);

        this.mvc.perform(request).andExpect(checkStatus).andExpect(checkBody);
    }

    @Test
    void testGetByPlatform() throws Exception {
        RequestBuilder request = get("/availableOn/Netflix");

        List<Movie> testGetByPlatform = List.of(new Movie(1, "Lord Of The Rings", "Adventure", 2001, "Netflix"));
        String testGetByPlatformAsJSON = this.mapper.writeValueAsString(testGetByPlatform);
        ResultMatcher checkStatus = status().isOk();
        ResultMatcher checkBody = content().json(testGetByPlatformAsJSON);

        this.mvc.perform(request).andExpect(checkStatus).andExpect(checkBody);
    }

    @Test
    void testGetByMovieName() throws Exception {
        RequestBuilder request = get("/getByName/Lord Of The Rings");

        Movie testGetByMovieName = new Movie(1, "Lord Of The Rings", "Adventure", 2001, "Netflix");
        String testGetByMovieNameAsJSON = this.mapper.writeValueAsString(testGetByMovieName);
        ResultMatcher checkStatus = status().isOk();
        ResultMatcher checkBody = content().json(testGetByMovieNameAsJSON);

        this.mvc.perform(request).andExpect(checkStatus).andExpect(checkBody);
    }

}

```

Spring-movies (15 Dec 2021 20:51:13)

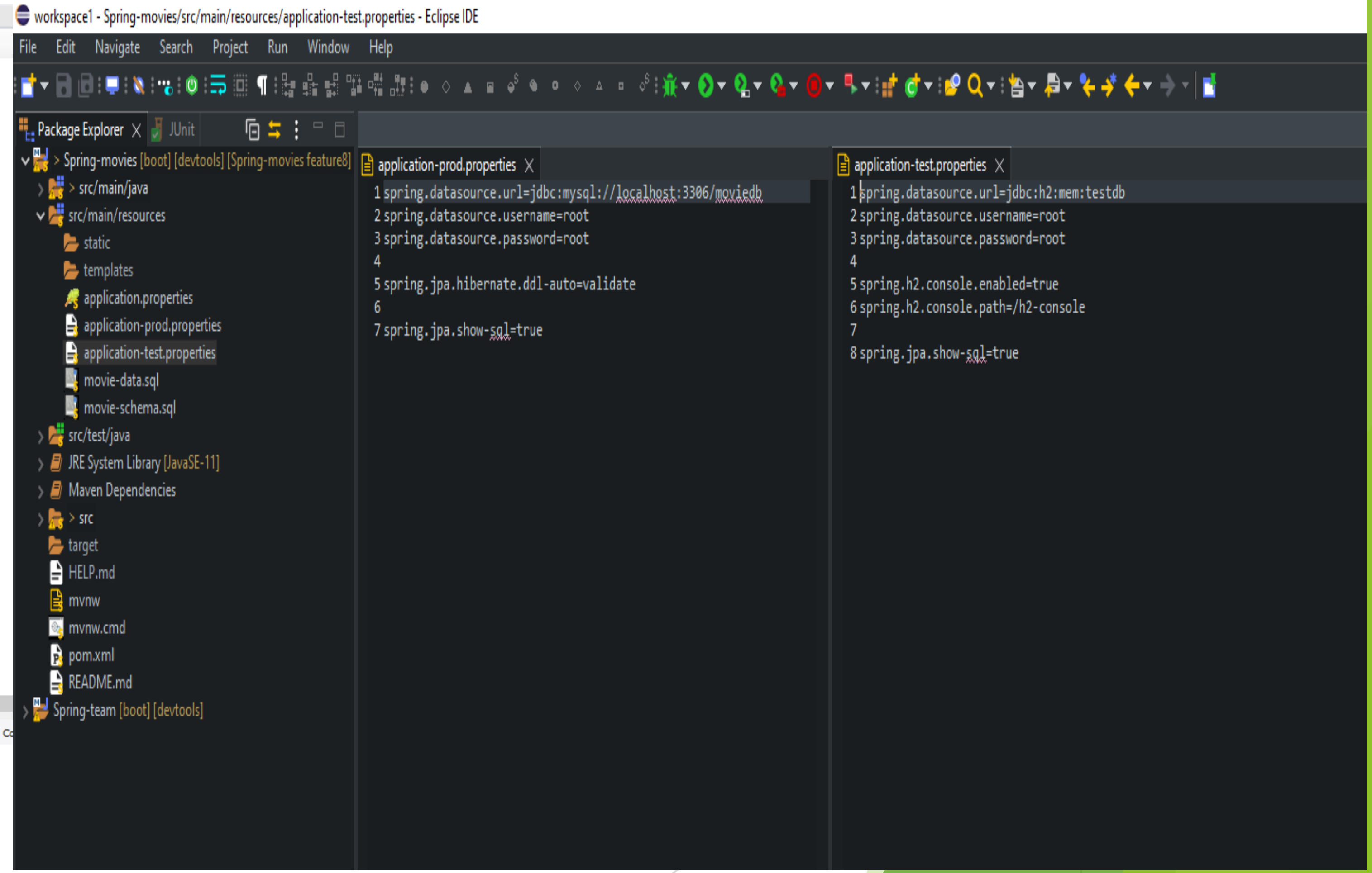
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Spring-movies	93.9 %	550	36	586
src/main/java	85.9 %	220	36	256
com.bae.movies.domain	64.4 %	56	31	87
com.bae.movies	37.5 %	3	5	8
com.bae.movies.service	100.0 %	82	0	82
com.bae.movies.web	100.0 %	79	0	79
src/test/java	100.0 %	330	0	330
com.bae.movies	100.0 %	4	0	4
com.bae.movies.web	100.0 %	326	0	326

SQL Schema and connection properties



```sql
DROP SCHEMA `moviedb`;

CREATE SCHEMA IF NOT EXISTS `moviedb`;
USE `moviedb`;

CREATE TABLE IF NOT EXISTS `movie` (
    `id` INT(11) NOT NULL AUTO_INCREMENT,
    `movie_name` VARCHAR(40) DEFAULT NULL,
    `genre` VARCHAR(40) DEFAULT NULL,
    `year_released`INT(11) DEFAULT NULL,
    `available_on` VARCHAR(40) DEFAULT NULL,
    PRIMARY KEY (`id`)
);
INSERT INTO `moviedb`.`movie`
(
`movie_name`,
`genre`,
`year_released`,
`available_on`)
VALUES
(
'Lord Of The Rings',
'Adventure',
2001,
'Netflix');
select*from movie;
```

id	movie_name	genre	year_released	available_on
1	The Lord Of...	Adve...	2001	Netflix
NULL	NULL	NULL	NULL	NULL


workspace1 - Spring-movies/src/main/resources/application-test.properties - Eclipse IDE
File
Edit
Navigate
Search
Project
Run
Window
Help
Package Explorer
JUnit
Spring-movies [boot] [devtools] [Spring-movies feature8]
src/main/java
src/main/resources
static
templates
application.properties
application-prod.properties
application-test.properties
movie-data.sql
movie-schema.sql
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
README.md
Spring-team [boot] [devtools]
application-prod.properties
application-test.properties

application-prod.properties

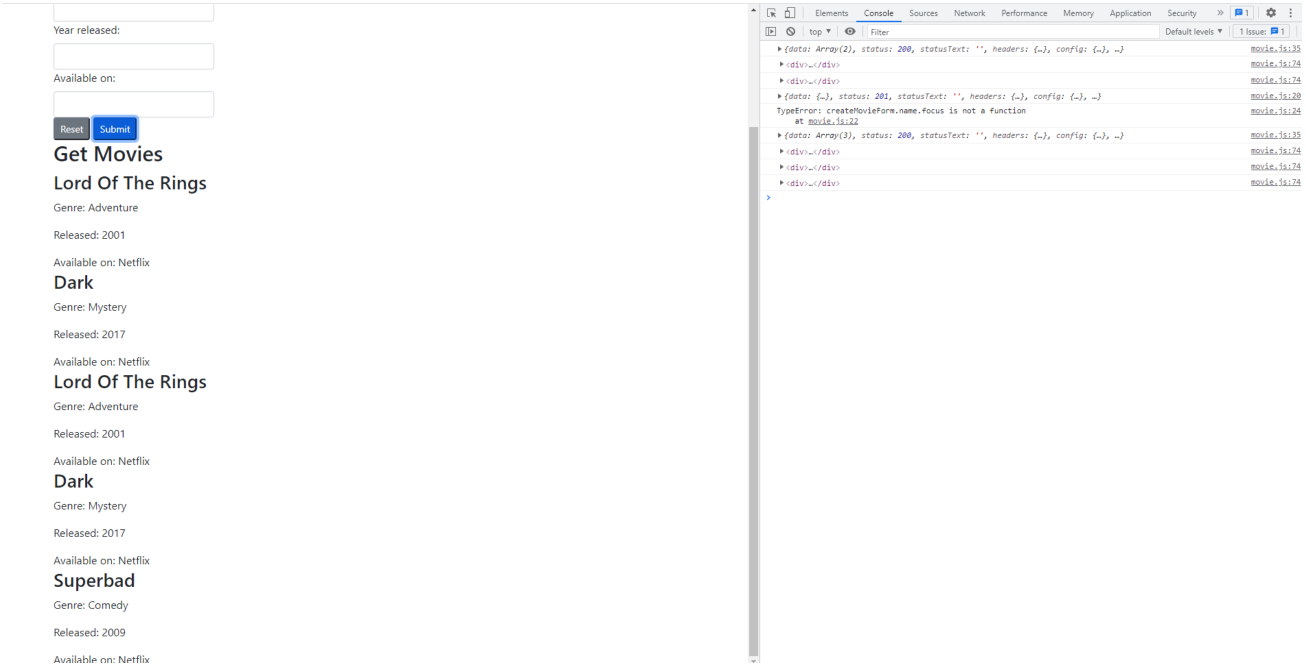
```
spring.datasource.url=jdbc:mysql://localhost:3306/moviedb
spring.datasource.username=root
spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=validate

spring.jpa.show-sql=true
```

application-test.properties

```
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.username=root
spring.datasource.password=root

spring.h2.console.enabled=true
spring.h2.console.path=/h2-console

spring.jpa.show-sql=true
```

Some issues

Repeating values – fixed by adding blank innerHTML text.



Year released:

Available on:

Get Movies

Lord Of The Rings

Genre: Adventure

Released: 2001

Available on: Netflix

Dark

Genre: Mystery

Released: 2017

Available on: Netflix

Lord Of The Rings

Genre: Adventure

Released: 2001

Available on: Netflix

Dark

Genre: Mystery

Released: 2017

Available on: Netflix

Superbad

Genre: Comedy

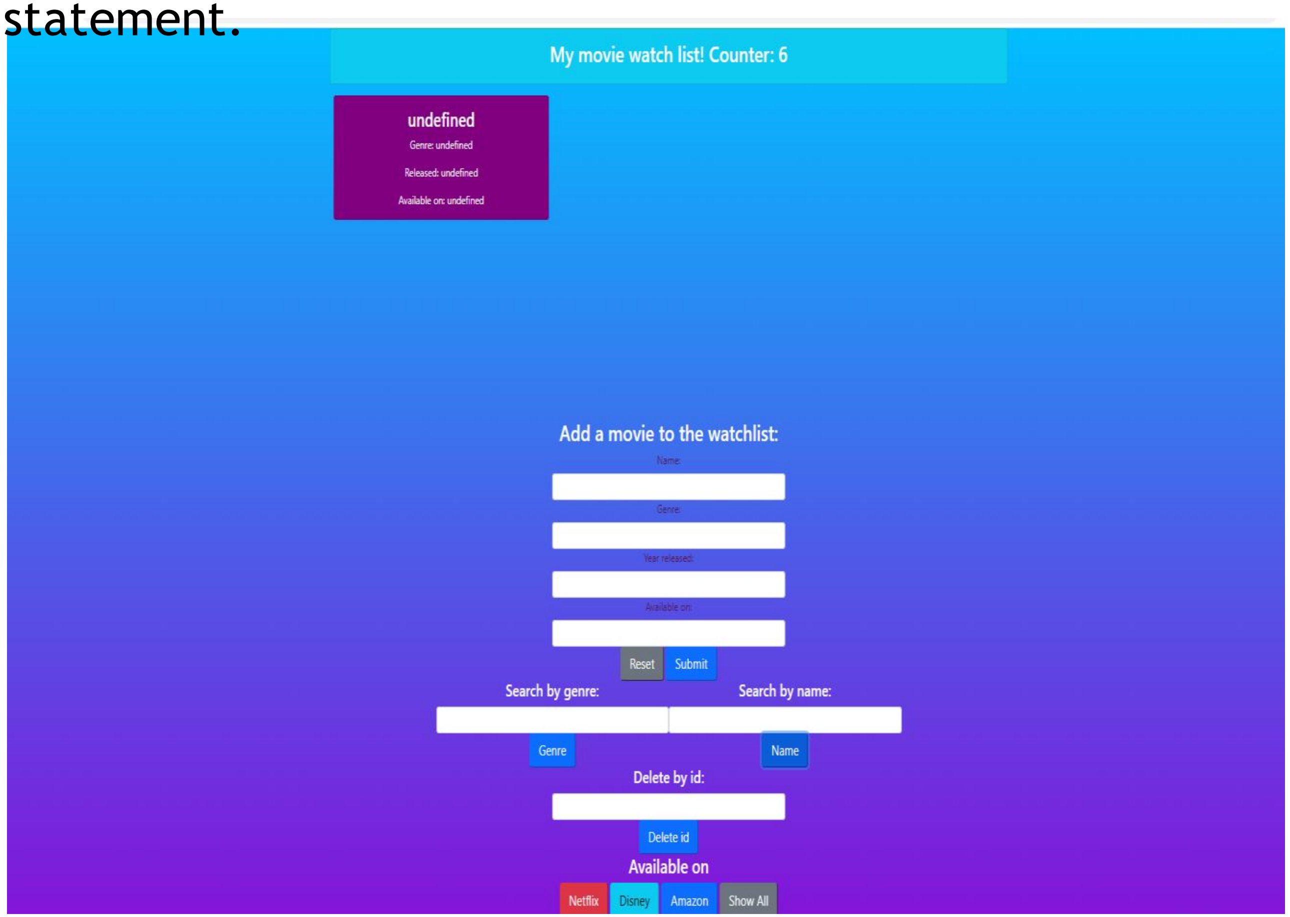
Released: 2009

Available on: Netflix

```
▸ {data: Array(2), status: 200, statusText: '', headers: {…}, config: {…}, …}    movie.js:35
▸ <div>…</div>                                                                    movie.js:74
▸ <div>…</div>                                                                    movie.js:74
▸ {data: {…}, status: 201, statusText: '', headers: {…}, config: {…}, …}          movie.js:20
TypeError: createMovieForm.name.focus is not a function                           movie.js:24
    at movie.js:22
▸ {data: Array(3), status: 200, statusText: '', headers: {…}, config: {…}, …}    movie.js:35
▸ <div>…</div>                                                                    movie.js:74
▸ <div>…</div>                                                                    movie.js:74
▸ <div>…</div>                                                                    movie.js:74
>
```

Issues cont…

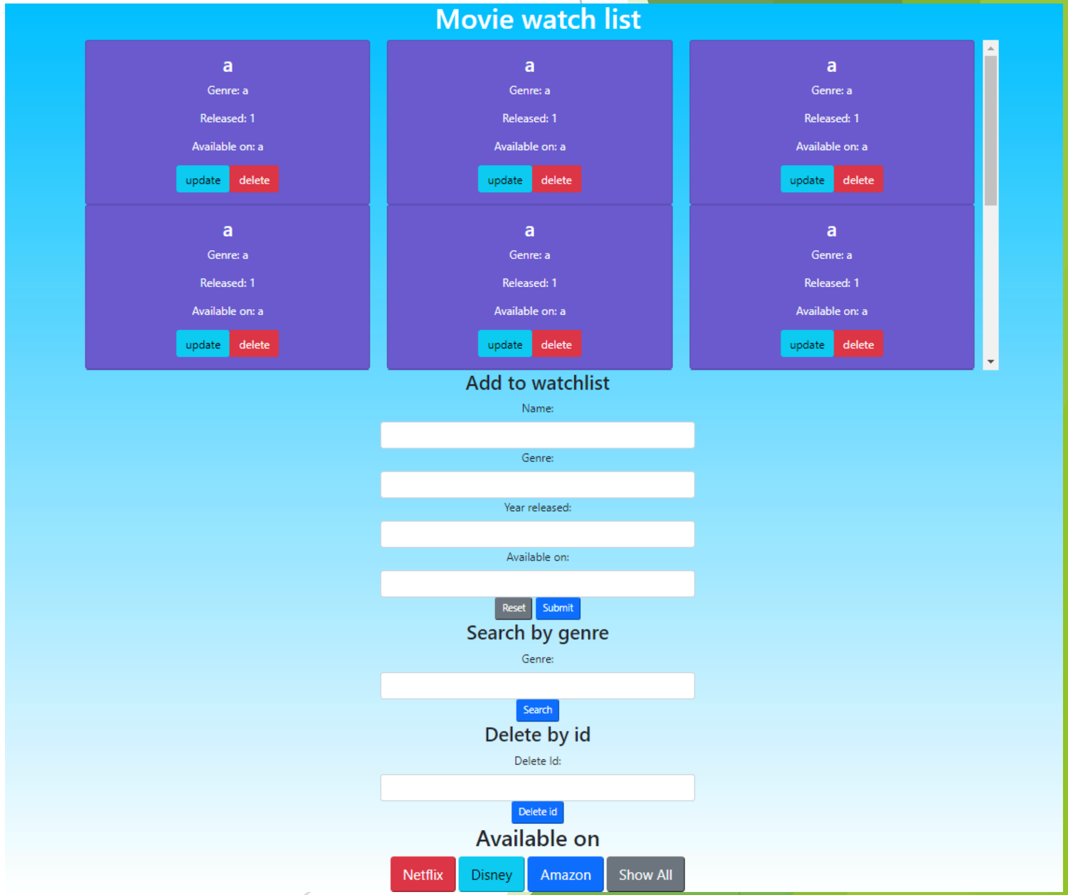
Undefined issue when name not in db – fixed by adding If statement.



My movie watch list! Counter: 6

undefined
Genre: undefined
Released: undefined
Available on: undefined

Repeating updates



Movie watch list

a
Genre: a
Released: 1
Available on: a

a
Genre: a
Released: 1
Available on: a

a
Genre: a
Released: 1
Available on: a

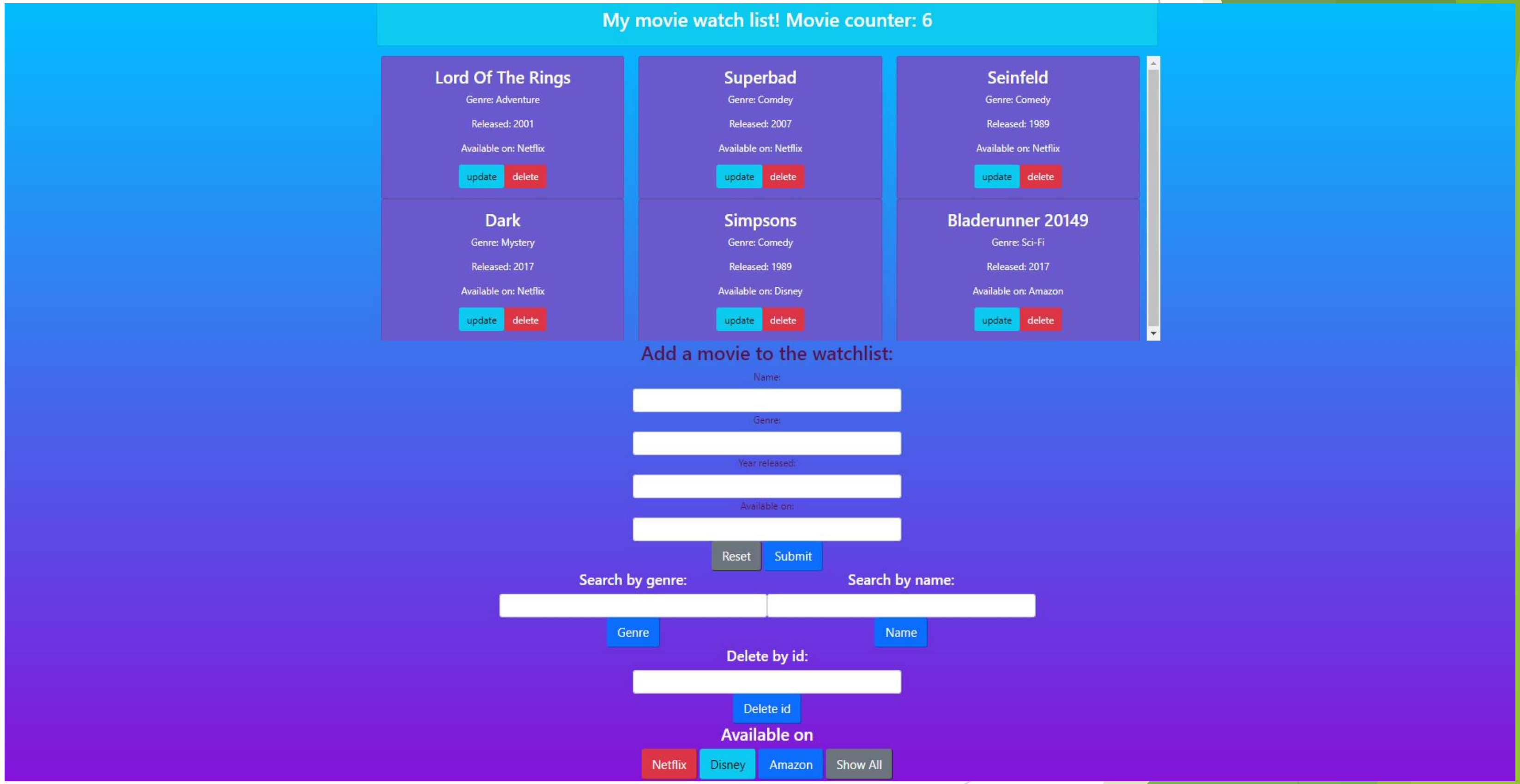
a
Genre: a
Released: 1
Available on: a

a
Genre: a
Released: 1
Available on: a

a
Genre: a
Released: 1
Available on: a

Hobby project demonstration

- READ ME
- MVP
- Demonstration



My movie watch list! Movie counter: 6

Lord Of The Rings
Genre: Adventure
Released: 2001
Available on: Netflix

Superbad
Genre: Comdey
Released: 2007
Available on: Netflix

Seinfeld
Genre: Comedy
Released: 1989
Available on: Netflix

Dark
Genre: Mystery
Released: 2017
Available on: Netflix

Simpsons
Genre: Comedy
Released: 1989
Available on: Disney

Bladerunner 20149
Genre: Sci-Fi
Released: 2017
Available on: Amazon

Add a movie to the watchlist:

Name:

Genre:

Year released:

Available on:

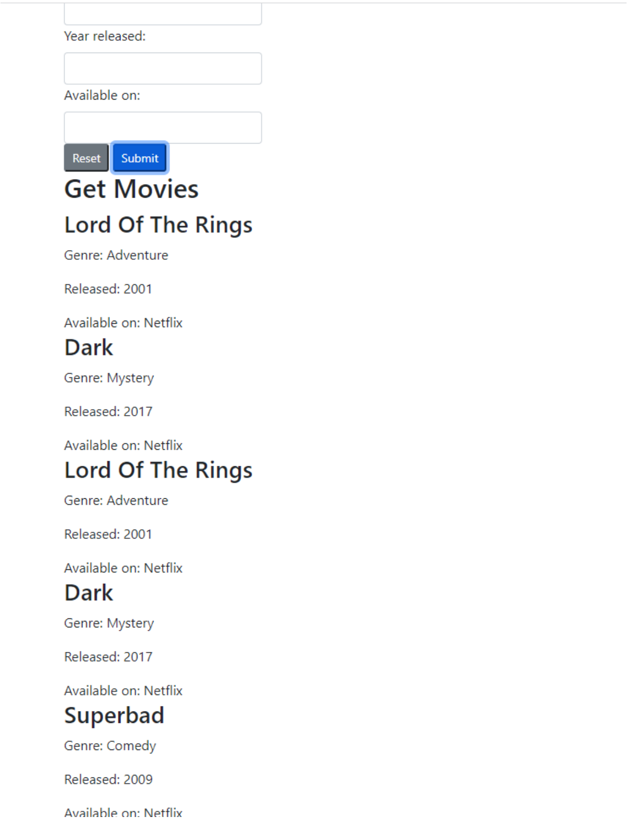
Search by genre:

Search by name:

Delete by id:

Available on

Webpage progression



Year released:

Available on:

Get Movies

Lord Of The Rings

Genre: Adventure

Released: 2001

Available on: Netflix

Dark

Genre: Mystery

Released: 2017

Available on: Netflix

Lord Of The Rings

Genre: Adventure

Released: 2001

Available on: Netflix

Dark

Genre: Mystery

Released: 2017

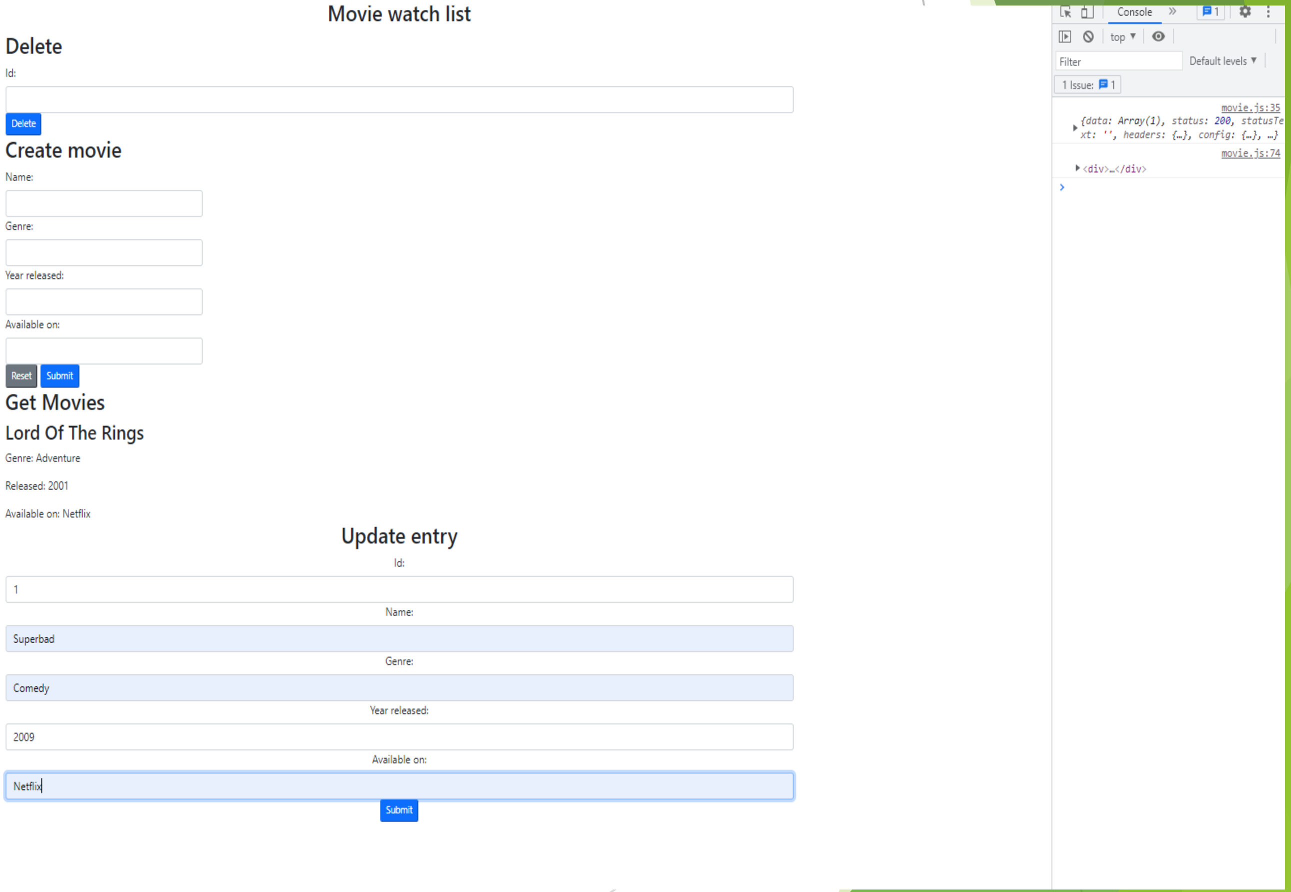
Available on: Netflix

Superbad

Genre: Comedy

Released: 2009

Available on: Netflix



Movie watch list

Delete

Id:

Create movie

Name:

Genre:

Year released:

Available on:

Get Movies

Lord Of The Rings

Genre: Adventure

Released: 2001

Available on: Netflix

Update entry

Id: 1

Name: Superbad

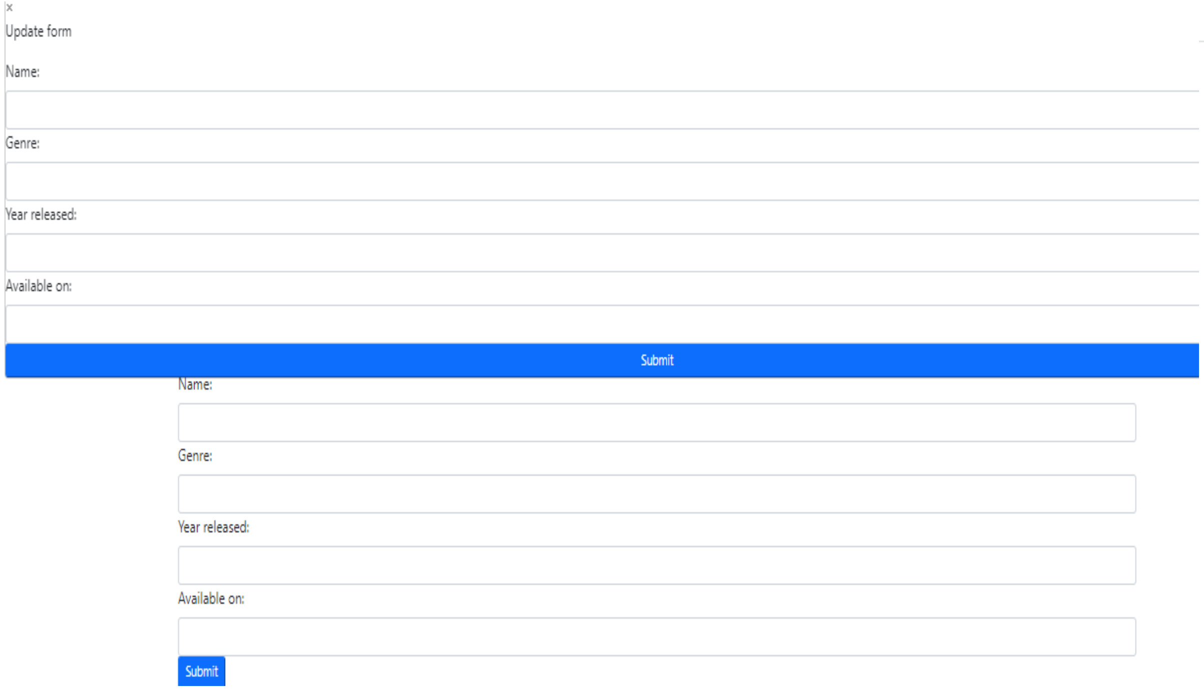
Genre: Comedy

Year released: 2009

Available on: Netflix

```
movie.js:35
{data: Array(1), status: 200, statusTe
xt: '', headers: {…}, config: {…}, …}
movie.js:74
<div>…</div>
>
```

Webpage progression



x

Update form

Name:

Genre:

Year released:

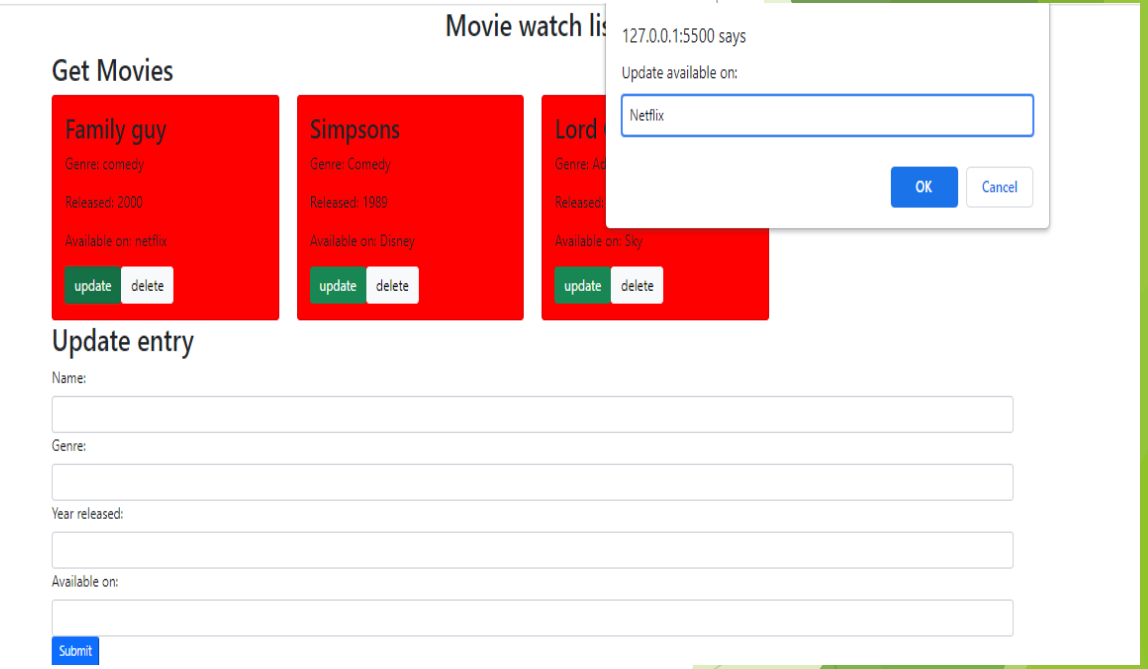
Available on:

Name:

Genre:

Year released:

Available on:



Movie watch lis

Get Movies

Family guy

Genre: comedy

Released: 2000

Available on: netflix

Simpsons

Genre: Comedy

Released: 1989

Available on: Disney

Lord

Genre: Ad

Released:

Available on: Sky

127.0.0.1:5500 says

Update available on:

Netflix

Update entry

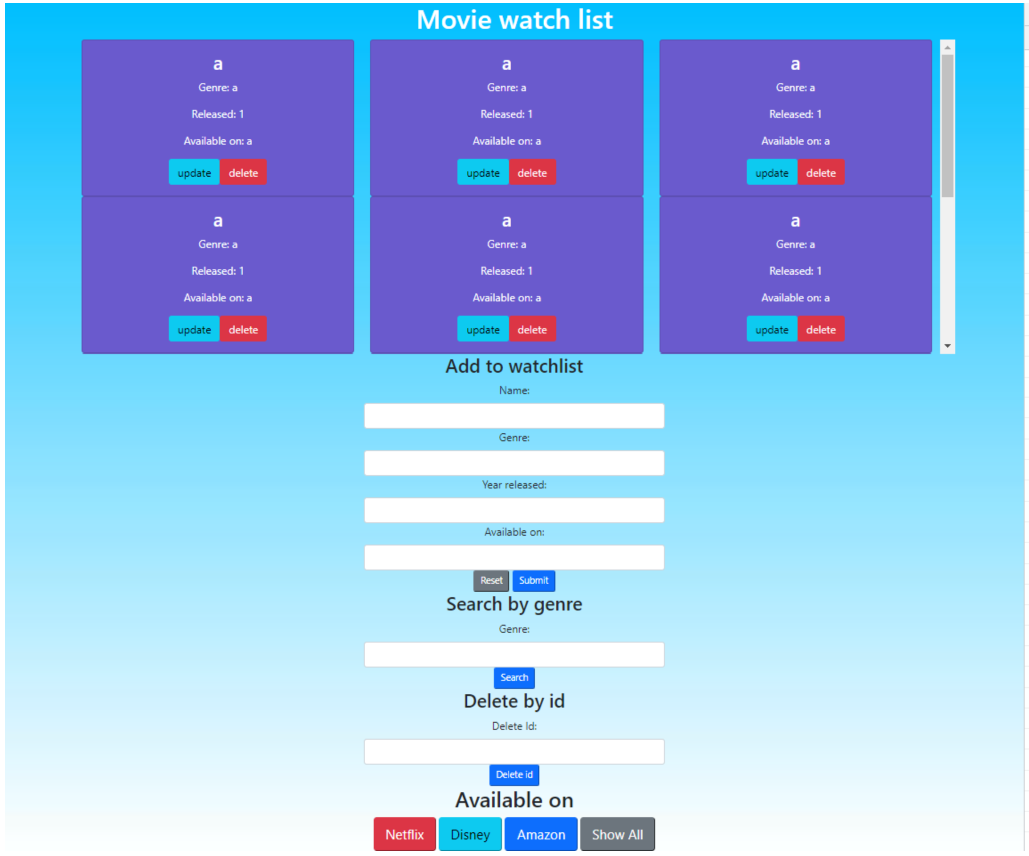
Name:

Genre:

Year released:

Available on:

Webpage progression



Movie watch list

a
Genre: a
Released: 1
Available on: a

a
Genre: a
Released: 1
Available on: a

a
Genre: a
Released: 1
Available on: a

a
Genre: a
Released: 1
Available on: a

a
Genre: a
Released: 1
Available on: a

a
Genre: a
Released: 1
Available on: a

Add to watchlist

Name:

Genre:

Year released:

Available on:

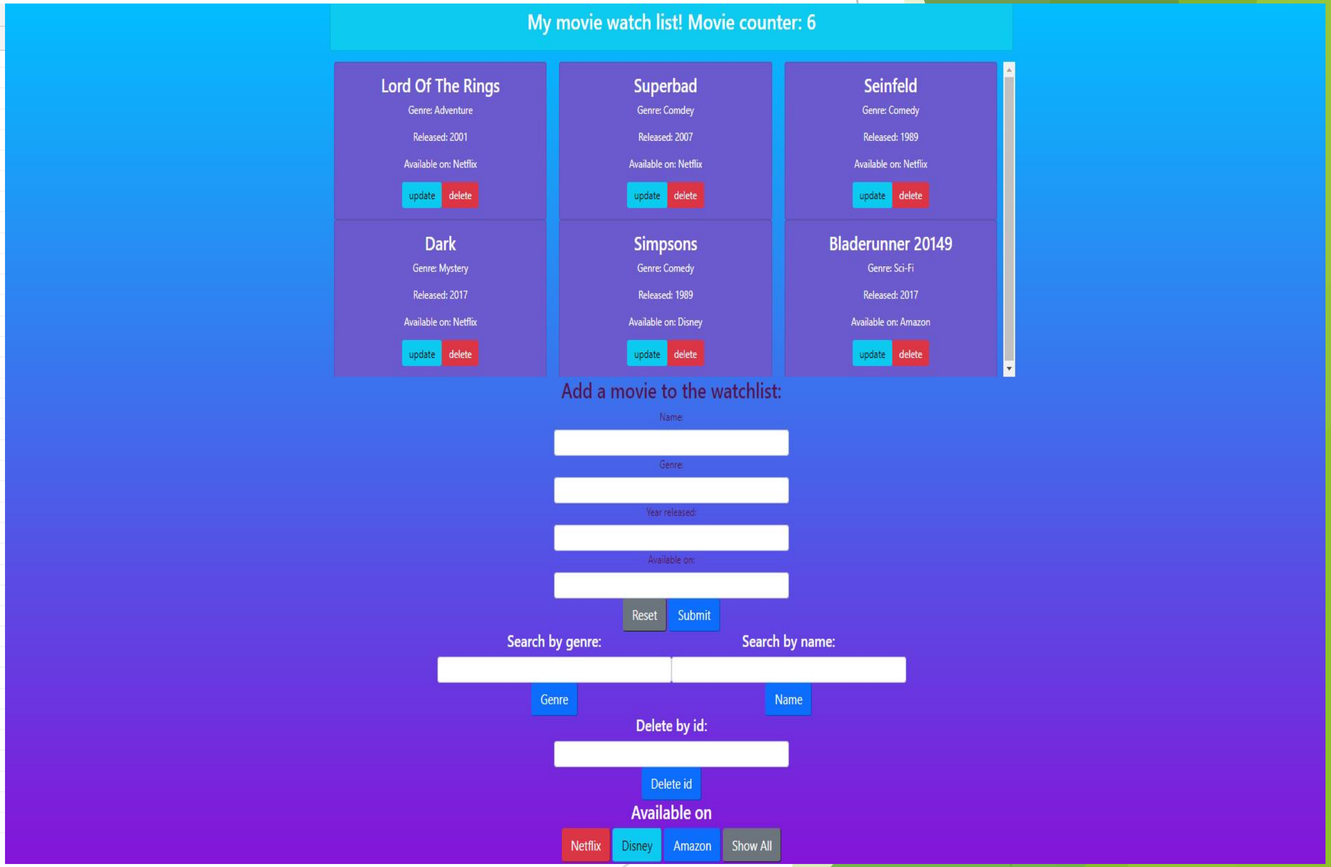
Search by genre

Genre:

Delete by id

Delete Id:

Available on



My movie watch list! Movie counter: 6

Lord Of The Rings
Genre: Adventure
Released: 2001
Available on: Netflix

Superbad
Genre: Comdey
Released: 2007
Available on: Netflix

Seinfeld
Genre: Comedy
Released: 1989
Available on: Netflix

Dark
Genre: Mystery
Released: 2017
Available on: Netflix

Simpsons
Genre: Comedy
Released: 1989
Available on: Disney

Bladerunner 20149
Genre: Sci-Fi
Released: 2017
Available on: Amazon

Add a movie to the watchlist:

Name:

Genre:

Year released:

Available on:

Search by genre:

Search by name:

Delete by id:

Available on

Sprint review

- Completed?

  Worked at a good pace.

  Did not need much assistance.

  Learnt a lot about the technologies used.

  Functioning application that meets the MVP and more.

- What got left behind?

  QOL of life improvements.

  Extra testing not in the MVP.

Sprint retrospective

- What went well?
- Time management, motivation, commitment, performed better than I thought.

- What could go better next time?
- Emphasis on planning – much better understanding of how projects connect and how powerful planning could be.
- Had a hard time testing.

Conclusion and reflections

- Overall successful happy with result.
- Very interesting project showing how applications use multiple technologies to achieve a goal.
- Greater appreciation for Git usage – see on webpages.
- Building on my skills and learning new technologies.

- Future steps.
- Project offered a strong opportunity for development and will continue to build my skills more along the apprenticeship.
- Learnt a lot of new technologies in a short period and happy with progress + expanded on previous technologies learnt.

Thank you

- Any questions?