Rattigan's Reviews

Team Three

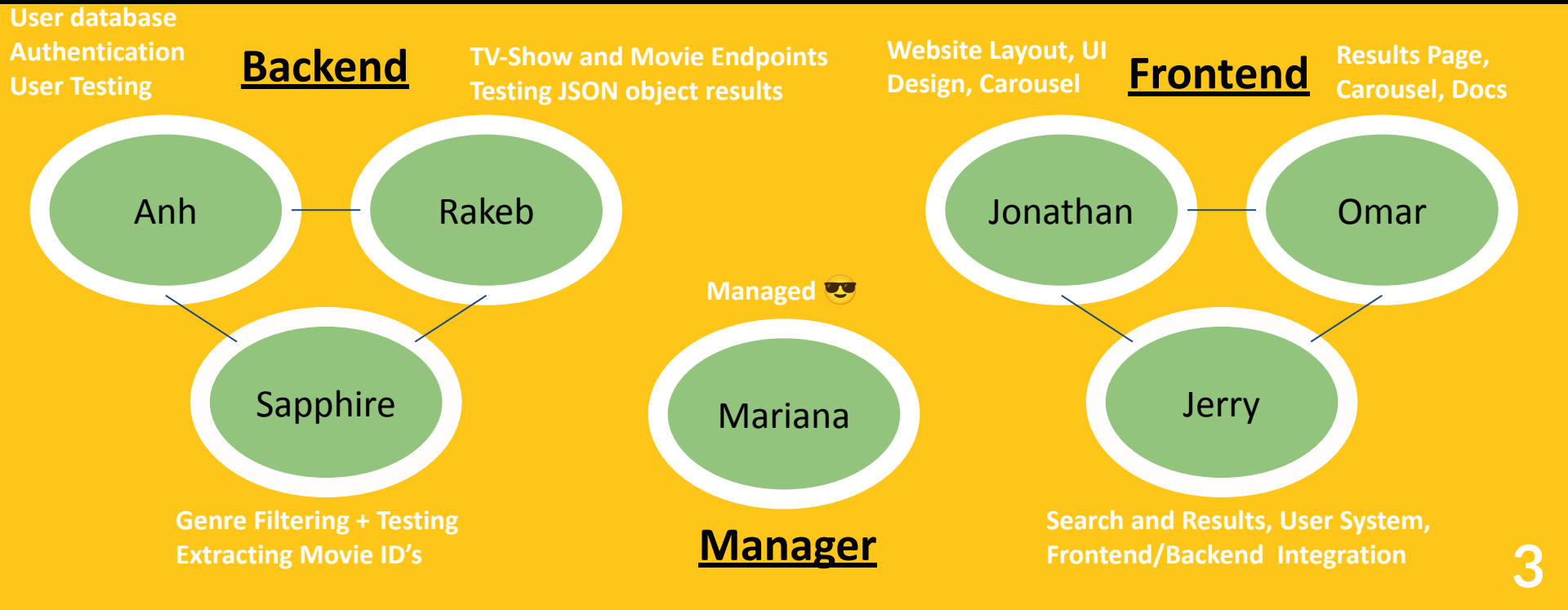
Presenting Today:

Omar, Sapphire, Rakeb

Agenda

- Members of the team & organization
- Project overview & Main Features
  - Architecture Diagram + Website Diagram
  - Tech Stack
- Testing
- Demo
- Project Retrospective and Future Outlook

Members of the Team & Organization



Team organization

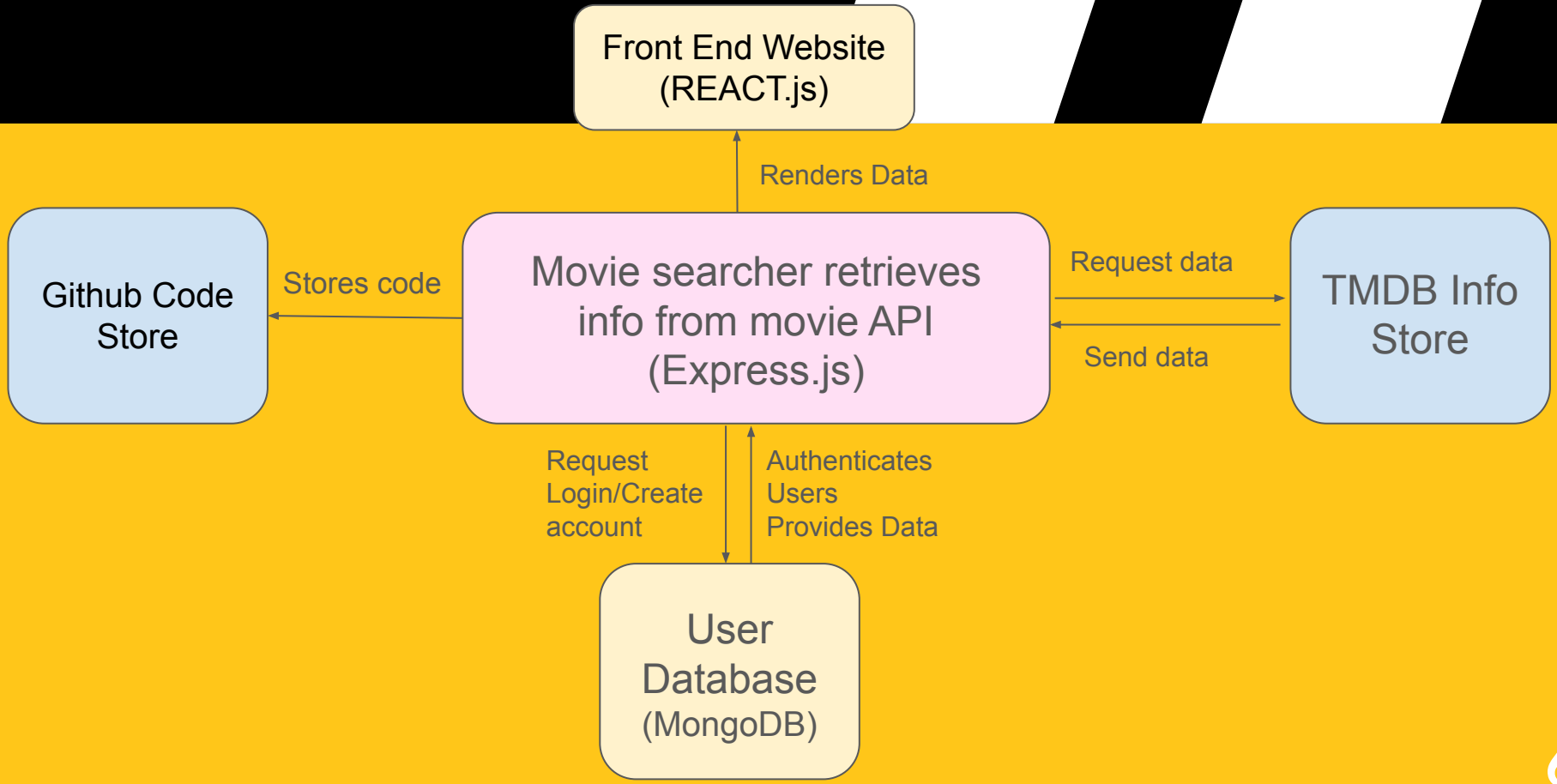
- Team was split into two groups of three, one working on frontend and the other on backend
- Trello to track project progress and goals/features
- Github to centralize work and keep version history
- Discord was our main form of communication

Key Features

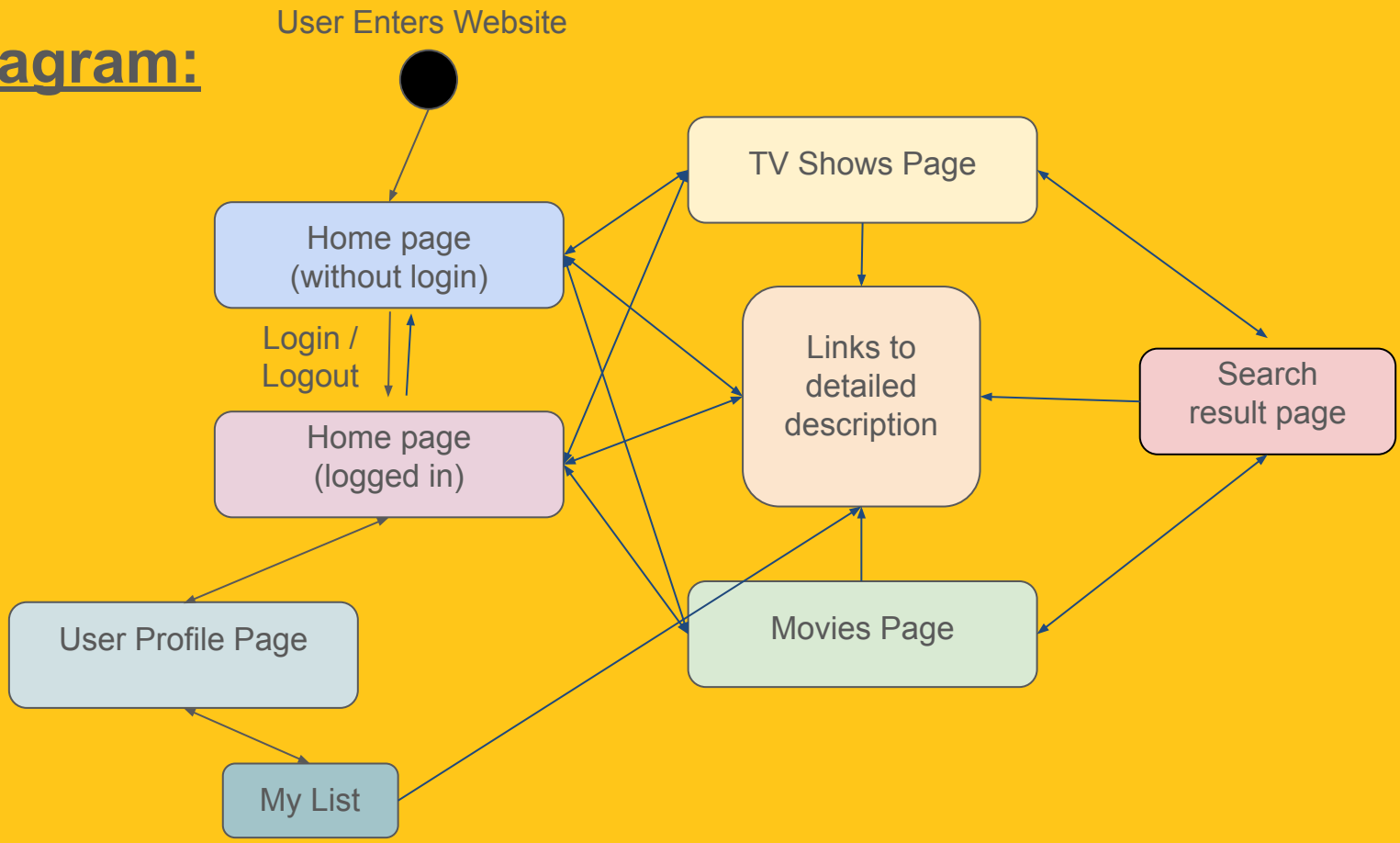
Rattigan's Review is a website designed to help all users find TV shows and movies, using a variety of criteria

The website features scrollbars with recommendations and a filtered search bar allowing users to search up shows/movies based on titles or genres.

Architecture Diagram:

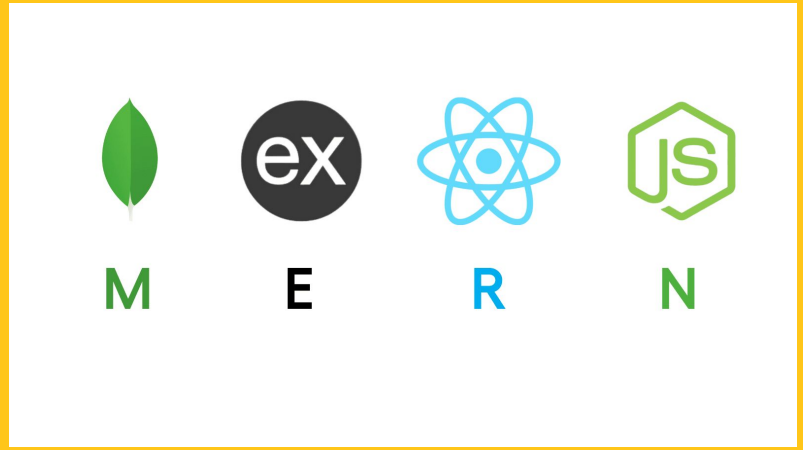


Website Diagram:



Tech Stack

- **Frontend:** React.js and Material UI
- **Backend:** MongoDB, Express.js, Node.js



Testing

```
Test Suites: 1 passed, 1 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        2.489 s
Ran all test suites.
```

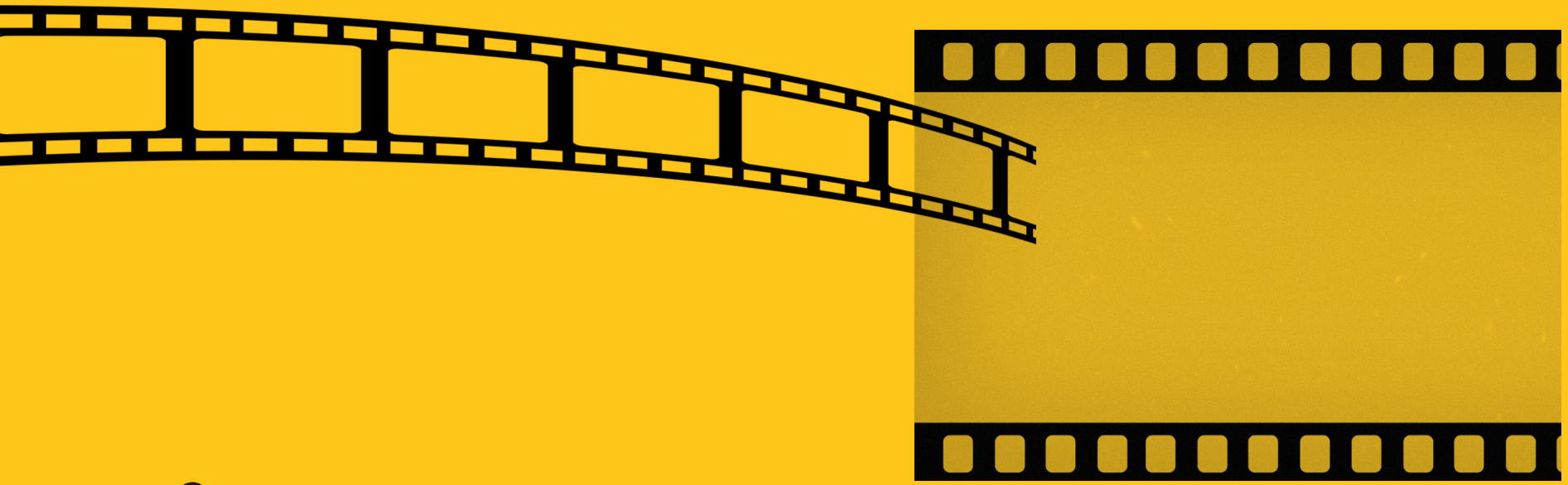
- **Frontend:**
  - Mainly manual testing features to make sure they work
  - Cypress testing to replicate user actions.
- **Backend:**
  - Manual testing endpoints on Postman.
  - Jest tests for handling search queries (keys-words and genres, async functions calling upon TMDB API), user profile tests(signup, login, logout,user duplication)

Testing Code

```
const request = require('supertest');
const { app } = require('../app.js'); // Ensure this points to your Express app


const sum = require("./sum"); //sum is just for testing purposes

const {sC, connectTV, connectMovie} = require('../app.js');
const splitCinema = sC;
const TMDBConnectionTV = connectTV;
const TMDBConnectionMovie = connectMovie;

      Châu Anh Phạm, 3 days ago • New testing
test('splitCinema, space translates to %20', () =>{
    expect(splitCinema(" ")).toBe("%20");
});

test('break bad id', async () => {
    expect((await TMDBConnectionTV("breaking bad"))[0].id).toBe(1396);
});

test('salt id movie id should be 27576', async () => {
    expect((await TMDBConnectionMovie("salt"))[0].id).toBe(27576);
});
```

```
describe("Authentication API", () => {
    test("should sign up a new user", async () => {
        const response = await request(app)
            .post('/auth/signup')
            .send({ email: "testuser@example.com", password: "password123" });
        expect(response.statusCode).toBe(200);
        expect(response.body.message).toContain("User created successfully! 🥳");
    });

    test("should reject duplicate user registration", async () => {
        // First call to register the user
        await request(app)
            .post('/auth/signup')
            .send({ email: "testuser_jest@example.com", password: "password123" });

        // Second call should fail
        const response = await request(app)
            .post('/auth/signup')
            .send({ email: "testuser_jest@example.com", password: "password123" });
        expect(response.statusCode).toBe(500);
        expect(response.body.message).toContain("User already exists! Try logging in. 😀");
    });
```

```
    test("setGenres tv test", async () => {
        const response = await request(app)
        .post('/sendGenres')
        .send({genres:[], medium: "tv"});
        expect(Array.isArray(response.body));
        response.body.forEach((entry) => expect.objectContaining(
            {name: expect.any(String),      Magirush, 3 days ago • 2 t
            year:expect.any(String),
            description: expect.any(String),
            rating: expect.any(String),
            posterImage: expect.any(String)}));
    });
```



Final Demo

Retrospective

Useful Tools:

- Postman
- Jest

Biggest challenges

- Dependencies
- Database setup
- Jest set up

Done Differently

- Utilize Trello more for task management

Technical & organization skills learned:

- Scrum principles, utilizing testing software, working with APIs and node.js, backend and frontend interfacing, React and CSS

Undeveloped Features

Recommendations based on watched shows

Anime page and pirating links

With Another Semester

Friends, show/movie ratings and reviews

More complex search filtering - release date, director, etc.

Any Questions?

THANK YOU!