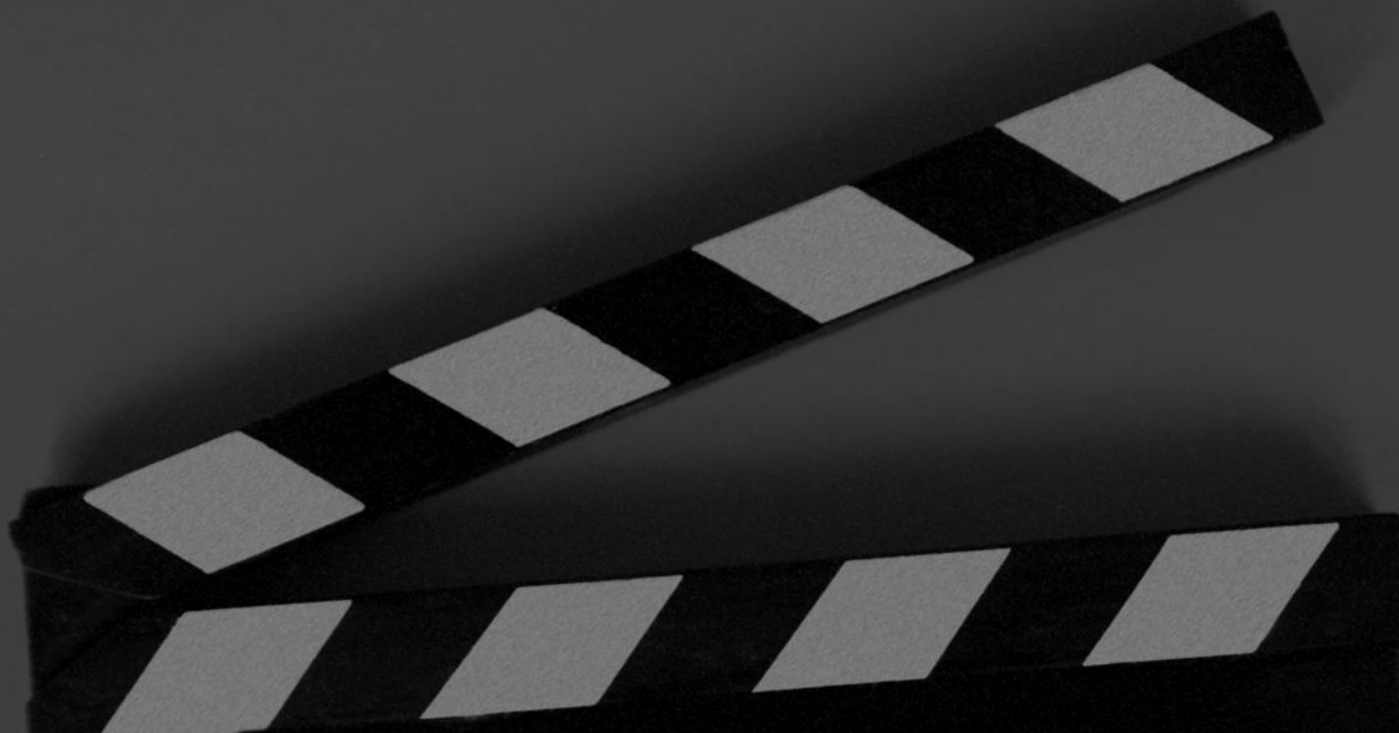


MovieApp

Group 4

Web programming application project



Team and Roles



- **Samuli Kokko** – fullstack development, architecture and project leading
 - User management, Finnkino showtimes, project documentation, application architecture, database design and publication in Render, Mocha tests and movie reviews
 - 245 hours
- **Janne Saarela** – fullstack development
 - Search movies, All groups, create groups feature, group pages add and delete movies to group, add showtimes to group, my info, Manage groups joinrequests, deletegroup, leave group
 - 111 hours
- **Nico Lappalainen** – fullstack development and project leading
 - Database creation, backend base models, base app layout sketch, favorite feature, favorite list feature, shareable favorite list feature, app UI design and app publication on Render
 - 110 hours
- **Mirella Lampinen** – fullstack development
 - Show groupmembers, delete groupmembers, Group page customization, moviecard component, responsiveness, light and dark mode
 - 115 hours

Technologies

Backend

- Node.js
- Express
- PostgreSQL (pg)
- JSON Web Token (jsonwebtoken)
- bcrypt
- Mocha & Chai (testing)
- dotenv

Frontend

- React (Vite)
- React Router DOM
- Axios
- React Hot Toast
- Font Awesome (FontAwesome)

Functionalities Checklist

| Status | Description |
|--------|---|
| ✓ | 1. Responsive design – Layout adapts to different screen sizes. |
| ✓ | 2. User registration – Create an account with email and secure password. |
| ✓ | 3. Login & logout – Sign in and out with your user credentials. |
| ✓ | 4. Account deletion – Remove your account and related data. |
| ✓ | 5. Movie search – Search movies by at least three criteria. |
| ✓ | 6. Showtimes – Browse Finnkino theaters and movie schedules. |
| ✓ | 7. Group page – Create, view, and delete movie discussion groups. |
| ✓ | 8. Join requests – Send and approve group join requests. |
| ✓ | 9. Member removal – Owner can remove members; users can leave groups. |
| ✓ | 10. Group customization – Add movies and showtimes to your group page. |
| ✓ | 11. Movie reviews – Logged-in users can write and rate reviews (1–5 stars). |
| ✓ | 12. View reviews – All users can browse reviews and movie info. |
| ✓ | 13. Favorites list – Logged-in users can create their personal favorites list. |
| ✓ | 14. Share favorites – Share your favorites via a public link (URI). |
| ✓ | 15. Light and dark mode – User can switch between light and dark mode |
| ✓ | 16. Published application – Application published in Render |

Documentation & Testing

Tests

- ✓ Sign in
- ✓ Sign out
- ✓ Registration
- ✓ Account delete
- ✓ Reviews: get, delete and add

Documentation

- ✓ Readme.md
- ✓ API documentation
- ✓ Database diagram
- ✓ Scrum board
- ✓ UI sketches

Challenges and Solutions

User token handling

- Took a long time to get refresh token to work. New concept for us.
- We use a secure token-based system:
 - **Access token** stored in memory and sent in the **Authorization header**
 - **Refresh token** stored as an **HTTP-only cookie**
 - When the access token expires, a new one is issued automatically

Backend routes and controllers

Code flow and multiple files handling

React component handling

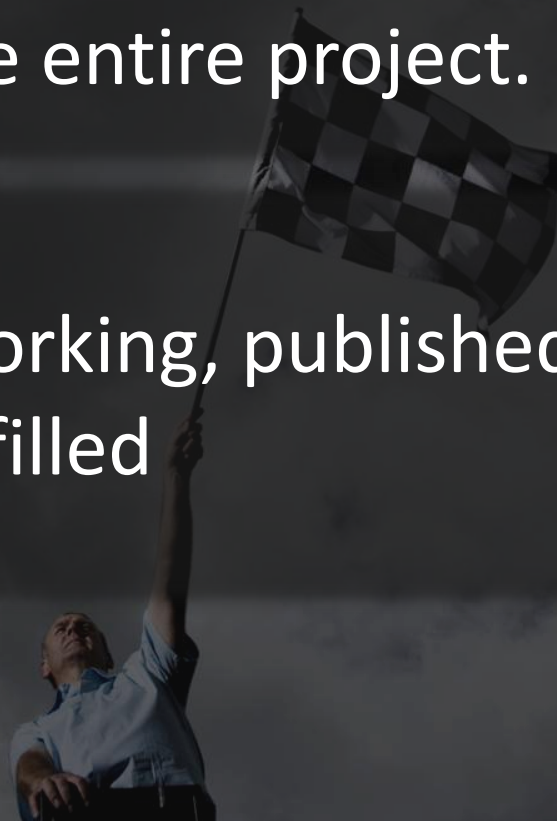
- for example, favorite and group buttons

Learnings

- How to build a full-stack web application using **React, Node.js, Express, and PostgreSQL.**
- We practiced team collaboration and version control with GitHub
- We also improved our problem-solving and debugging skills through real project challenges
- Right now, we are improving our presentation skills in English

Successes

- Team work worked well. Everyone was present in every meeting and remained active during the entire project.
- Application is working, published and every requirement fulfilled



Future Improvements

- Refining the user experience and UI
- Implementing more testing
- Improving code reliability and error proofing
- Refactoring the code
- After that, new features:
 - More social features



The End