



# QA-Cinema Group Project Presentation



# Contents

Introduction

Technologies Used

Planning

Team Organisation

Demo

Testing

Sprint Review

Project Evaluation

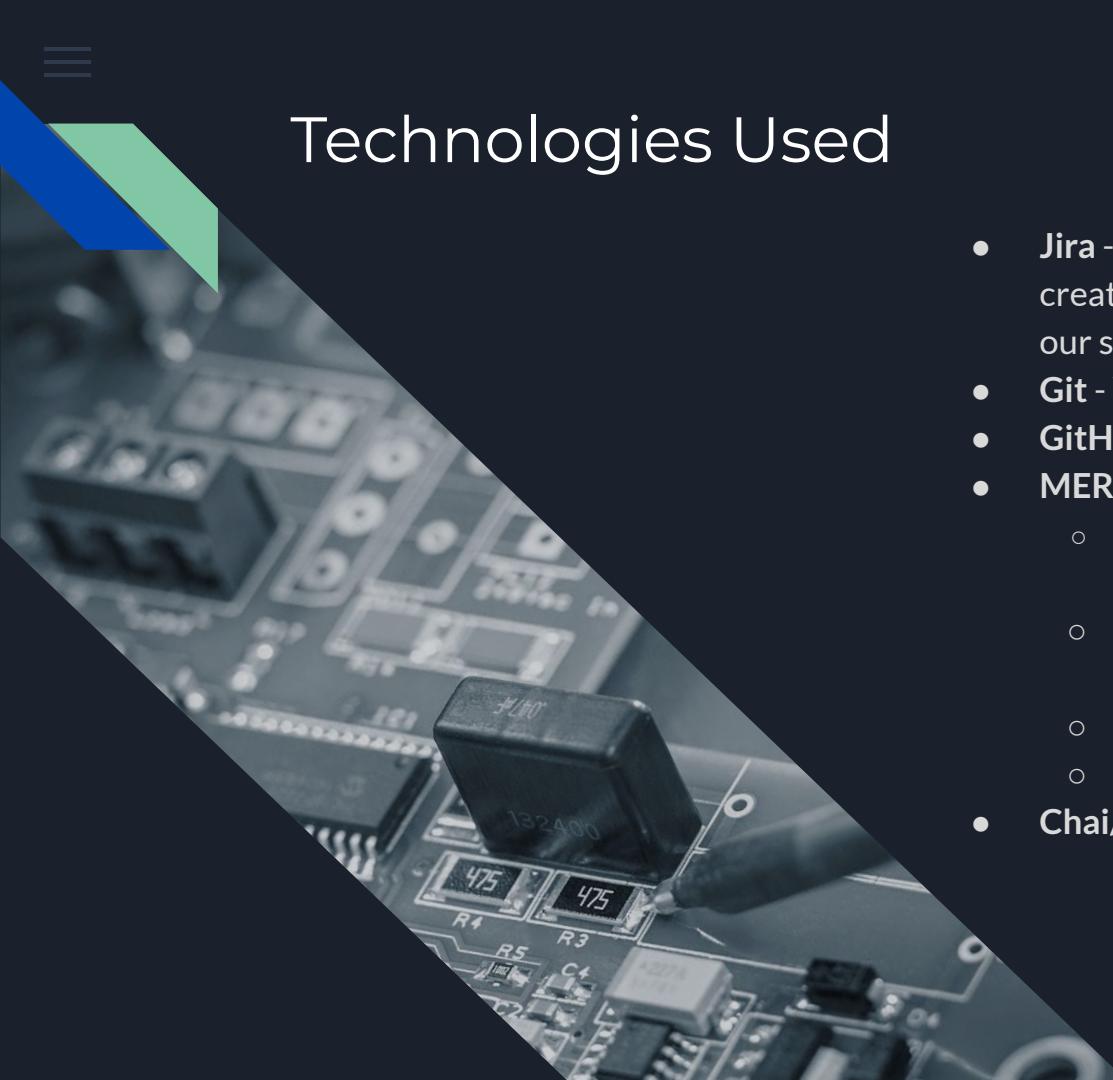
Future Functionality



# Introduction

For our group project, we have designed and built a web application for a cinema chain, QA Cinemas. Our objective was:

To create a full-stack Web application suitable for a given client specification, with utilisation of supporting tools, methodologies and technologies that encapsulate all modules covered during training



# Technologies Used

- **Jira** - The project management tool we used to create our product backlog and Kanban boards i.e our sprint as a part of our Scrum
- **Git** - Version control system
- **GitHub** - Source code management
- **MERN** stack
  - **MongoDB Atlas** - Our data management system hosted in the cloud.
  - **Express** - API Development Platform (Backend)
  - **React** - Front-End Development
  - **Node** - Back-End Development
- **Chai/Mocha** - Unit Testing



# Planning

- 
- 01 Jira - Decided on user stories and created sprints for our project.
  - 02 Risk Assessment - Created a list of risks that could have occurred while completing the project.
  - 03 Table Structure - Created tables for the back-end of our web application where the data will be stored.
  - 04 Project Setup - Setting up MongoDB and a group GitHub repo.
  - 05 Wireframes - Creating templates for our web pages.

# Planning - Jira

- Assigned a Scrum Master, Stephanie.
- Created user stories and built a product backlog.
- Assigned stories points.
- Created sprints with Kanban boards.
- Planned daily stand ups and sprint retrospectives.

The screenshot shows the Jira Software interface with a dark theme. At the top, there's a navigation bar with links for 'Jira Software', 'Your work', 'Projects', 'Filters', 'Dashboards', 'People', 'Apps', and a 'Create' button. The main area is titled 'QA-cinema Software project' and 'QC Sprint 1'. It displays a Kanban board with three columns: 'TO DO 26 ISSUES', 'IN PROGRESS', and 'DONE'. The 'TO DO' column contains several user stories, each with a title, a 'GENERAL' label, and a checked checkbox labeled 'QC-39', 'QC-60', 'QC-37', 'QC-41', 'QC-3', and 'QC-2'. The 'IN PROGRESS' and 'DONE' columns are currently empty. On the left side, there's a sidebar with sections for 'PLANNING' (Roadmap, Backlog, Board), 'DEVELOPMENT' (Code), 'OPERATIONS' (Deployments), and 'Project pages', 'Add shortcut', and 'Project settings'. A message at the bottom left says 'You're in a team-managed project' with a 'Learn more' link.

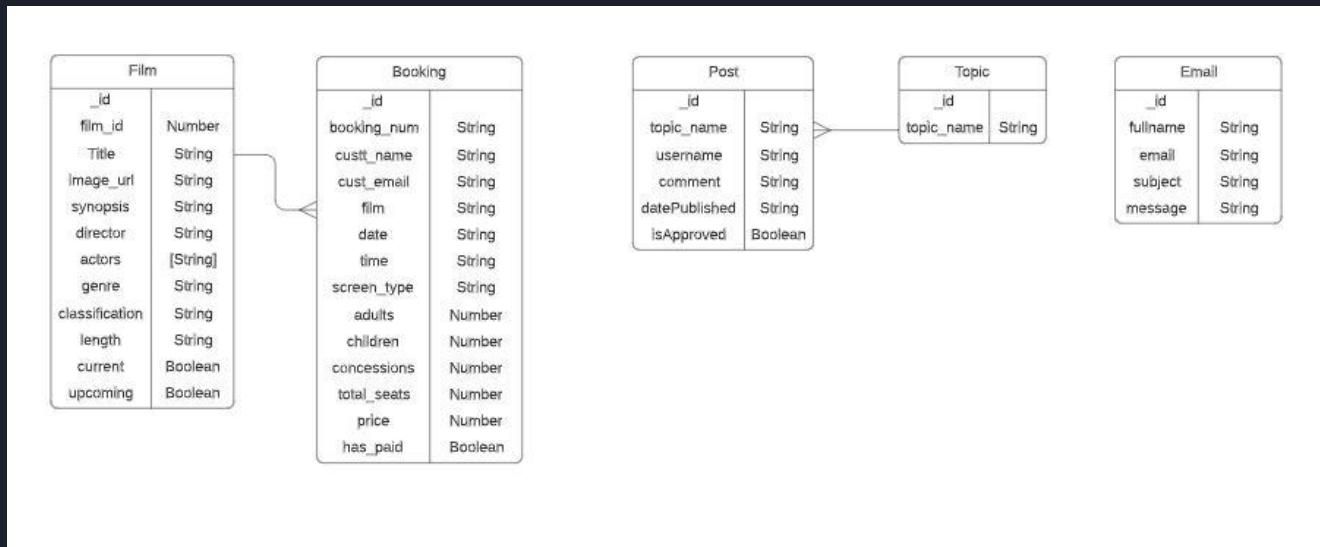
# Planning - Risk Assessment

Created a list of risks and how we may deal with them if they occur. Wrote control measures to reduce the chance of them occurring. Assigned a risk level.

Risk	Statement	Response	Objective	Control Measures	Likelihood	Impact	Risk Level
Internet outage	Loss of access to Jira, GitHub, Teams, Mongo Atlas	Continue to work locally, making local commits. Use alternate locations for internet e.g. libraries, internet cafes	To keep the project on schedule and minimise delays.	Look into internet alternatives beforehand incase internet does go down.	Unlikely	Moderate	Yellow
Team member falls ill	Unable to continue working and contribute their part of the work	Inform trainer and rest of the team members. Reassign the tasks to other team members.	To ensure all tasks can still be completed.	Have a premade contingency plan. Avoid burnout by taking breaks. Eat and sleep well.	Moderate	Moderate	Orange
Team members local machine stops functioning	Unable to continue working on the project and could potentially lose work.	Inform the team. Source a new machine and clone the github repository. Reassign tasks until the team member sorts out a new machine	To ensure minimal effect to the overall progress of the project	Push commits to github regularly. Scope out possible alternative machines or shops with fast delivery	Unlikely	Moderate	Yellow
Github server goes down	Issues pushing to github as the server is down.	Continue working on the project locally and push once the issue is resolved. Keep an eye on the github status updates. Ensure constant communication as a team so all team members are aware of the updates being made.	To ensure progress continues to be made. Constant communication to minimise merging errors when merges are eventually made on github.	Add commit and push to github regularly to ensure that the repository is as up-to-date as possible.	Unlikely	Moderate	Orange
Poor Risk Management	An unidentified risk occurs that could jeopardise the overall project	Identify potential risks thoroughly, calculating the likelihood of each and carefully monitoring risks.	To ensure there is minimal effect to team productivity should a risk occur	Identify potential risks thoroughly and ensure the likelihood of each risk is calculated and identified	Very Unlikely	Low	Green

# Planning - Table Structure

Designed a data structure diagram to plan how the data would be stored in the cloud database. This is where we started thinking about what data we need to actually process and store from the front-end



# Planning - Project Setup

Created a GitHub repository, created a folder for our API and React app. Installed dependencies. Setup a cloud database on MongoDB Atlas.

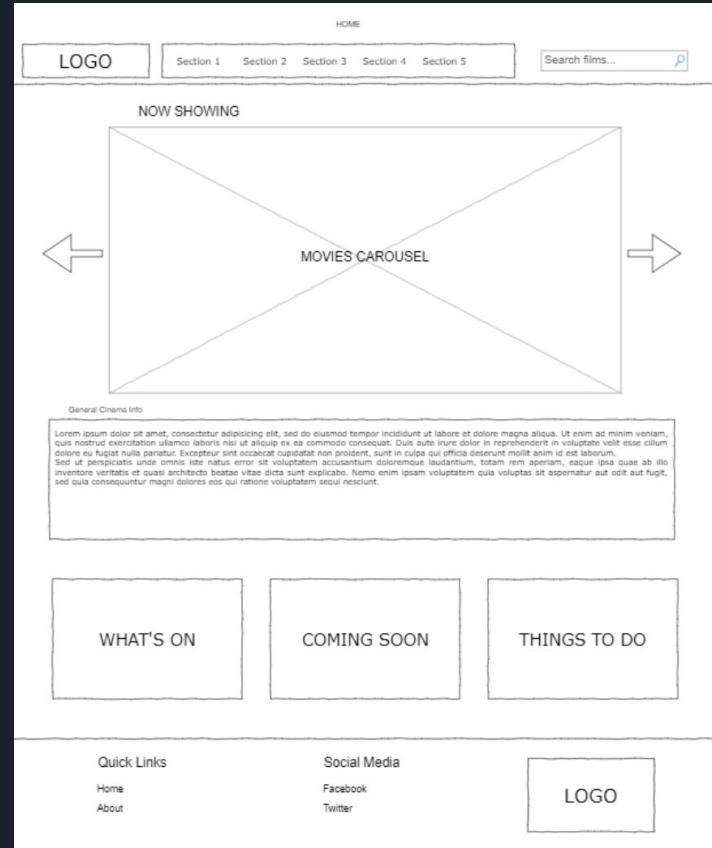
The screenshot shows the MongoDB Atlas interface for the 'QACinema' database. The left sidebar includes sections for Deployment (Database selected), Data Lake, PREVIEW, Data Services, Triggers, Data API, Data Federation, and Security (Database Access, Network Access, Advanced). The main area displays the 'KISHAN'S ORO - 2022-10-03 > PROJECT 0 > DATABASES' section for the 'qacinema' database. It shows 1 database and 6 collections. The 'Collections' tab is active, displaying a table with the following data:

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
bookings	11	3.12KB	291B	36KB	1	36KB	36KB
emails	5	859B	172B	36KB	1	36KB	36KB
films	8	5.7KB	730B	24KB	1	24KB	24KB
posts	22	4KB	187B	36KB	1	36KB	36KB
topics	11	687B	63B	36KB	1	36KB	36KB
uids	2	145B	73B	36KB	2	72KB	36KB

Buttons for 'CREATE COLLECTION' and 'REFRESH' are visible at the bottom right of the table.

# Planning - Wireframes

- Started thinking about the front end of the web application.
- Designed the layout of our pages to help guide our front end development using React.





# Team Organisation

Now that all the tasks were created on Jira and the initial set up was done, we started allocating tasks to members of the team.

Everyone had an opportunity to display their skills on both the front-end and back-end.

We then started the implementation for the project





Team Member 01



# Stephanie Ashdown

I enjoyed the challenge of working on this project and it was great for improving my skills particularly with React. I worked on the bookings system.

## Tasks include:

- Homepage
- Film Listings:
  - What's On
  - Upcoming
- Booking System
- Testing booking routes



Team Member 02

# Kishan Kunvardia

I enjoyed working on the front-end design but also liked the challenge of implementing the back-end. Working on the search functionality taught me a lot about React and Express.

Tasks include:

- Navigation Bar for all pages
- Search functionality
- Find Us
- Contact Us with Email form
- Classifications
- Collaborating on Discussions page
- Email form testing



Team Member 03

# Leo Gornovskiy

I enjoyed learning React for this project. I created a login system and worked on the footer.

Tasks include:

- Footer for all pages
- Custom logo creation
- About Us
- Login system for admins
- Login testing



Team Member 04

# Rebecca Swinton

I worked on both the frontend and backend for this project. Adding the functionality for the discussions page helped me understand React and Express a lot more .

Tasks include:

- Opening Times
- Places to Go
- Individual Film pages
- Collaborating on Discussion page
- Discussion testing



# Demo - QA Cinema





# Testing

Test DB connected

Booking tests

- ✓ Gets all booking
- ✓ Gets booking by booking number
- ✓ Creates a booking
- ✓ Gives error when create film without required field
- ✓ Updates a booking
- ✓ Deletes a booking

Email tests

- ✓ Gets all emails
- ✓ Gets one email
- ✓ Creates one email
- ✓ Create one email but required fields missing
- ✓ Delete all emails from a specific email address

Film tests

- ✓ Gets all current and future films
- ✓ Gets film by id
- ✓ Gets film with title included in search term
- ✓ Gets film with genre included in search term
- ✓ Gets all current films
- ✓ Gets all upcoming films
- ✓ Creates a film
- ✓ Gives error when create film without required field
- ✓ Updates a film
- ✓ Deletes a film

Post test

- ✓ Gets all Posts from discussion board
- ✓ Gets posts by topic name
- ✓ Gets unapproved posts
- ✓ Should create a new post
- ✓ Should not create a post without username field

Topic tests

- ✓ Gets all Topics from discussion board
- ✓ Should create a new topic
- ✓ Should not create a new topic

UID tests

- ✓ Gets one uid
- ✓ Creates one uid
- ✓ Create one uid but required fields missing

# Testing - Coverage

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	85.49	50	70.23	85.49	
node-express	95.83	50	100	95.83	
index.js	95.83	50	100	95.83	15
node-express/persistence/models	100	100	100	100	
Booking.js	100	100	100	100	
Email.js	100	100	100	100	
Film.js	100	100	100	100	
Post.js	100	100	100	100	
Topic.js	100	100	100	100	
UID.js	100	100	100	100	
node-express/routes	80.57	100	69.13	80.57	
UIDs.js	91.66	100	83.33	91.66	9
bookings.js	84.61	100	73.33	84.61	8,17,34,43
email.js	85.71	100	75	85.71	8,17,43
films.js	82.05	100	70.83	82.05	8,17,27,35,43,60,69
posts.js	63.33	100	50	63.33	8,17,33,38-42,48-52
topics.js	90.9	100	83.33	90.9	8



# Sprint Review

## What went well?

- Kept track of the tasks being completed by each team member
- Got through most of the tasks assigned for the sprints
- Achieved the MVP.

## What could be improved?

- Improve our story point estimation so we can distribute tasks more evenly.
- There were a few tasks that were incomplete
- Better task prioritisation, focusing on MVP tasks more than desirable requirements.



# Project Evaluation

## What went well?

- We all ran into issues but we were usually able to come together and solve them.
- Good communication
- Everyone willing to put in extra hours
- Planned and managed the project well.
- Everyone had an opportunity to display their front-end and back-end skills.

## What could be improved?

- More wireframing to support front-end development.
- Evenly distributing the workload.
- Implementing testing on the go instead of at the end
- Better coding practices.



# Future Functionality

- Implement a login system through the whole web application and all user types.
- Being able to login using other platforms e.g. LinkedIn, Google.
- Interface for admin to perform all CRUD functionality i.e add films to the site.
- Implement some front-end testing with Selenium.
- Allocating seats for a booking or adding a timer.
- Creating a user profile/dashboard.
- Displaying trailers for films.



Thank you!

Any questions?

