Bookero – Project Report

Project Vision:

Bookero aims on becoming one of the most sophisticated online bookstore on the market today.
Our mission is to provide quality but affordable books for education, entertainment, self-development while providing a wide range of books to satisfy any customer's expectations. We aim to exceed our customer's expectation while giving shop owners a lot of freedom on what they want to sell and how they want to sell. We are essentially acting agents of trust among the customer, the publisher and the shop owner, while we work on more ambitious tasks keeping the trust on hand of these three parties remain our topset priority.

Bookero plans to implement complex features which are going to use Artificial Intelligence and Machine Learning in the future which will enable us to recommend custom books to every user on their preferences. This will make the Bookero user experience truly personal for every user.

Bookero is starting with it's operations in Australia, then plan to expand to the APAC region and as the portal grows we plan to provide accessible knowledge around the globe.

Our vision is to make education as accessible as possible even to the most remote places on the map.

System Architecture:
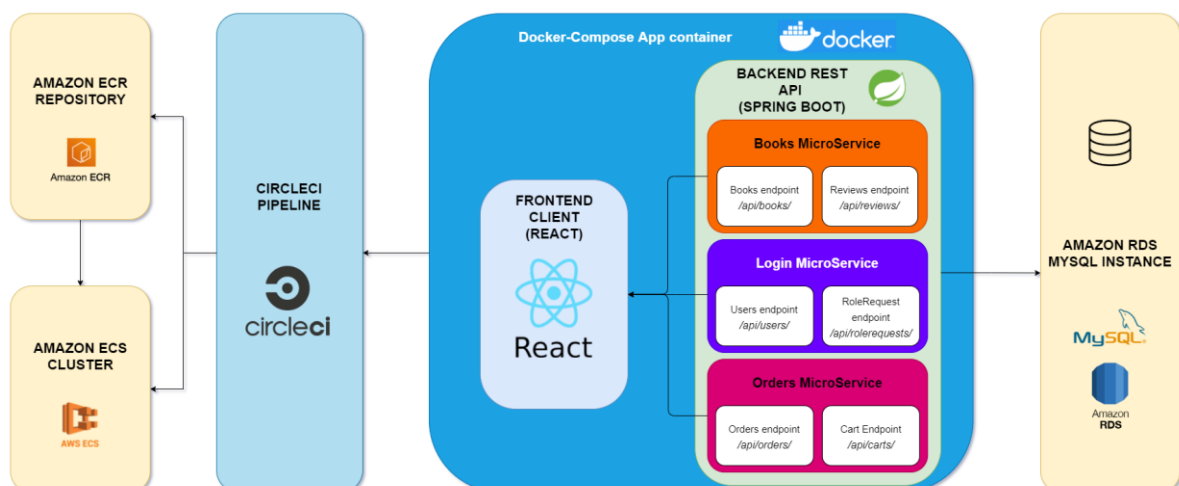
GitHub workflow:

Out GitHub workflow strategy was implemented from the first sprint itself.
We used a TDD approach to develop everything, our GitHub workflow was as follows :

- Master branch is the production branch where all the CI/CD is configured
- Develop branch is the staging branch where all the changes are merged into and tested before merging into Master branch
- To implement any new feature the developer creates a local branch on their machine which is branched from the Develop branch.
- This new branch show be pushed to upstream as well.
- Any experimental branch which the developer creates should only be kept on their local machine.
- The developers committed the code either just after completing the feature or if they have finished working for that day
- Merge to Develop and Master take place regularly
- Merges were done in developer meetings as any merge conflict faced could be resolved with absolute priority

Scrum Process:

Scrum Master: Kartik Kumar

Our Scrum Process were pretty well defined and well understood by the development team.

- Regular Stand-ups in the form of texts were done.
- The development team met every 3 – 4 days to discuss if there was any major change need to be implemented
- Effort meetings were conducted before every sprint
- Scrum Poker and User Stories discussions were highly regarded
- After every meeting, meeting minutes were noted down to keep a record and referring back
- Developers met almost everyday to discuss individual component of the codebase
- Regular meetings with the Product Owner to realise any change required.
- Dedicated meetings to reflect upon the Sprint Retrospective were held after every sprint

Refactoring Strategy:

Our refactoring strategy was really straight forward, this is how we refactored the code efficiently so that we ran into virtually no errors after completion and the project was created with industry standard code.

- Decoupling everything. We went and looked at every class and component and made sure that everything which can be unreferenced from one another was unreferenced. This was the key as if even one of the components break we wont have to worry about others.
- Removal of any Magic Numbers
- Removal of any unwanted comments
- Writing required comments
- Renaming variables to be highly self-descriptive
- Renaming functions to the agreed standard in the initial plan of the project
- Making sure all dependencies are imported
- Cleaning up any unrequired files in the project structure

Suggested Enhancement:

Bookero team is looking forward to make this product the best and have a flawless User Experience. To achieve that we plan to work on these enhancements after Alpha test release.

- We plan to do a User Experience test survey of the portal to see where the possible gaps are.
- Security implementations can be improved, the application can be deployed on a VPC (virtual private cloud)
- The app could use a write protected database inside RDS
- Strict CORS configuration is recommended
- AWS WAF (web application firewall) is recommended to be used to prevent from any kind of attack
- The application should be built on AWS Cloud Front so that we can use the Edge Locations to ensure low latency to the end users
- S3 bucket objects can have a strict bucket policy rather than being publicly accessible.
- Further the app can benefit from GraphQL for more complex backend operations.

CircleCI Deployment Pipeline:

TEST PHASE

BUILD PHASE

DEPLOY PHASE

Build Bookservice Image
- Creates docker image for books microservice and pushes the image onto a private AWS ECR repository tagging the image as latest for ECS to later deploy it

Build Loginservice Image
- Creates docker image for login microservice and pushes the image onto a private AWS ECR repository tagging the image as latest for ECS to later deploy it

TEST
- Runs docker-compose on a circleCI virtual machine to ensure the application compiles and passes pre-written deployment tests

Build Orderservice Image
- Creates docker image for orders microservice and pushes the image onto a private AWS ECR repository tagging the image as latest for ECS to later deploy it

Deploy Development/Production
- Depending on the branch (either development or master respectively) the built images are launched to the development or production ECS cluster overwriting the previous deployments image

Build frontend Image
- Creates docker image for frontend Client service and pushes the image onto a private AWS ECR repository tagging the image as latest for ECS to later deploy it