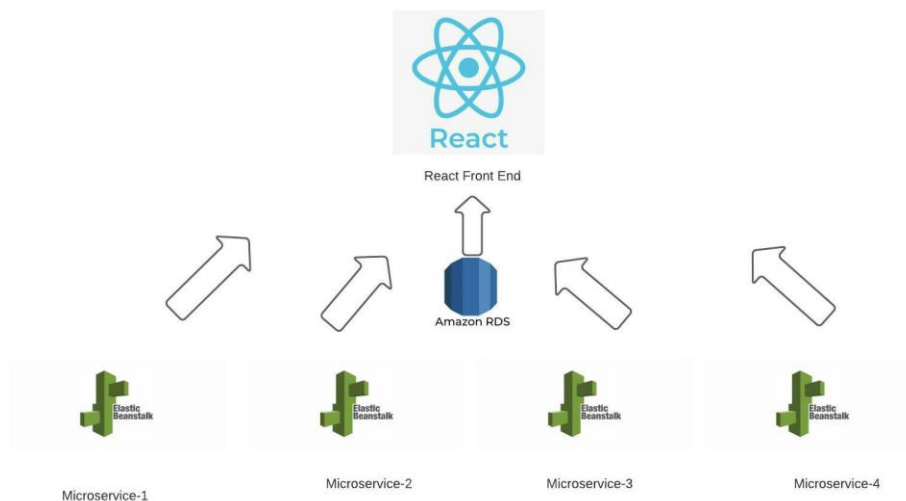


PROJECT REPORT (GROUP 3 TUES1830- REZA)

Vision Statement

To provide a user-friendly app that will allow for all majority of ages to navigate through and find books they have an interest in purchasing. Being the best means giving our users functionalities that they can confidently use and understand in order to search for a book depending on any attribute they are interested in. They not only can search for books easily, they can place an order through PayPal to buy the book, request a refund within 2 hours of placing the order, view their order status and transaction history as well as place a review for the book they purchased. Building a product with little harm to user is an important vision for us as we believe every user should have data integrity and proper security. This possible as the admin users constantly approve/deny user registrations, look out for harmful/disrespectful reviews placed, able to edit/remove books published if seemed inappropriate and are able to have access to getting transactions history to help with any enquiries. Not only do we have a safe app to navigate through, we are also allowing publishers/shop owners to publish their hard worked books onto our app for customers to be able to purchase. Publishers/Shop owners are able to see history of transaction orders, status of current orders and change the availability of a book. This BOOKERO app will make it very simple and secure for a user to navigate through the app, making it an enjoyable experience that no person should miss out on using if you love reading books.

Basic System Architecture (Front, backend datastore)



Refactoring Project

Down below you will find parts of the project where we changed/enhanced in order to improve the functionality of our app and ensure we are adhering to the assignment specifications.

Set orders to appear from most recent to least recent:

```
@GetMapping("/findOrders")
public List<Order> findOrders() {
    List<Order> returnList = orderService.getOrders();
    Collections.reverse(returnList);
    return returnList;
}
```

Now the purpose for this refactoring was that user looks to find the order then can get the list from most recent to make it easier to look for the order rather than needing to try find them in an unordered list.

Created calendar instance to better work with SQL date format:

```
@GetMapping("/findOrdersForRefund/{username}")
public List<Order> findOrdersForRefund(@PathVariable String username) {
    Calendar c = Calendar.getInstance();
    Date referenceDate = new Date();
    c.setTime(referenceDate);
    c.add(Calendar.HOUR, amount: -2);
    Date date = c.getTime();

    List<Order> returnList = orderService.getOrdersForRefund(date, username);
    Collections.reverse(returnList);
    return returnList;
}
```

Now the purpose for this refactoring was to make the format for SQL to work better for our instance to ensure efficiency of our code.

Added Response entity to signify report creation success or failure:

```
    ResponseEntity<Object> responseEntity = ResponseEntity.ok().headers(headers)
        .contentType(MediaType.parseMediaType("application/txt")).body(resource);

    return responseEntity;
} catch (IOException e) {
    e.printStackTrace();
    return new ResponseEntity(HttpStatus.BAD_GATEWAY);
}
```

Now the purpose for this refactoring was to test to see if report has been created properly or if it hasn't been created properly.

Allowed refunds for orders that are of a specific status:

```
public List<Order> getOrdersForRefund(Date date, String username) {
    List<Order> returnList = orderRepository.findAllByCreateAtAfterAndUsernameAndStatus(date, username, status: "Order Placed");
    returnList.addAll(orderRepository.findAllByCreateAtAfterAndUsernameAndStatus(date, username, status: "Shipped"));
    return returnList;
}
```

Now the purpose for this refactoring was that refunds can only occur for an order that has been placed rather than having refunds occur for someone that hasn't even placed an order. So, this checks to see the specific status before giving a refund.

Replaced removed redundant fields in book table (category removed) and

Added pdf_link to make it available to download:

```

@Entity
@Table(name = "BOOK_TBL")
public class Book {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @NotBlank(message = "ISBN is required")
    @Column(unique = true)
    private String ISBN;

    @NotBlank(message = "Title is required")
    @Column(unique = true)
    private String title;

    @NotBlank(message = "Author's name is required")
    private String author;

    @NotBlank(message = "Genre is required")
    private String genre;

    @NotBlank(message = "Condition is required")
    private String condition;

    @NotBlank(message = "Price is required")
    private String price;

    @NotBlank(message = "Page Count is required")
    private int pageCount;

    @NotBlank(message = "Store Owner is required")
    private String storeOwnerID;
    private String pdf_link;
}

```

Now the purpose for this refactoring was that all unnecessary information is removed and so that information of the book is available for download as a PDF file which wasn't implemented properly before.

Set books to appear from most recent to least recent:

```

@GetMapping("/findBooks")
public List<Book> findBooks() {
    logger.trace("Retrieved all books");
    List<Book> returnList = bookService.getBooks();
    Collections.reverse(returnList);
    return returnList;
}

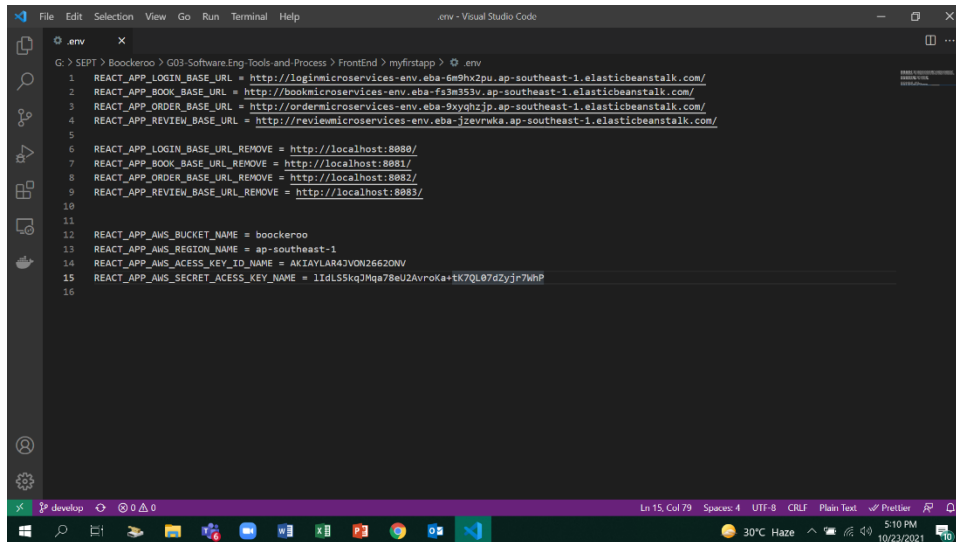
```

Now the purpose for this refactoring was that user can find books from most recent to least recent to see any new books that have been published on the app rather than trying to manually look for new books published.

Furthermore, general code cleanup like removing unnecessary imports, unnecessary type casts and removing unnecessary dependencies were also changed to enhance our project.

Front-End:

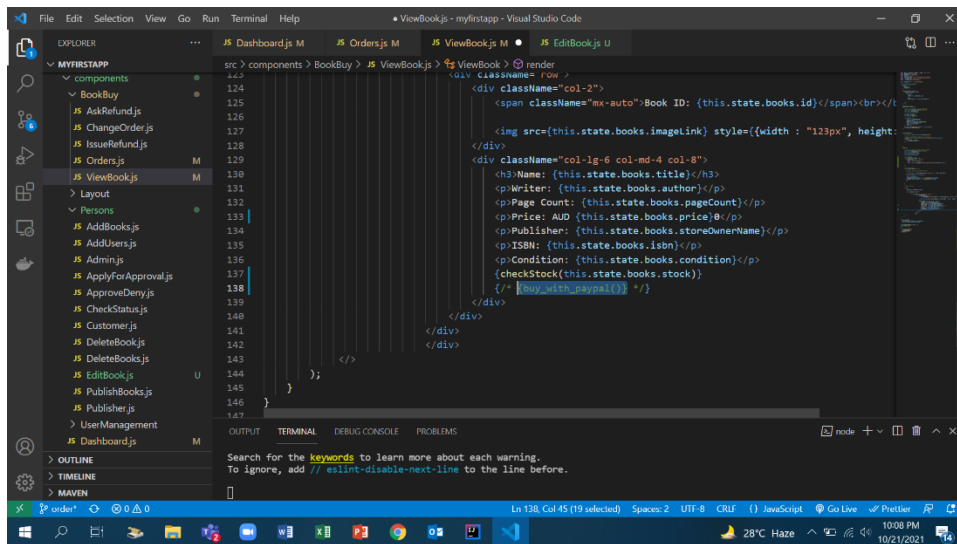
As a front-end developer, I had to dive through a lot of refactoring. One of the vulnerabilities of my code was that it was revealing all API credentials and AWS credentials in the JS file itself. Then I decided to put every credential in a .env file.



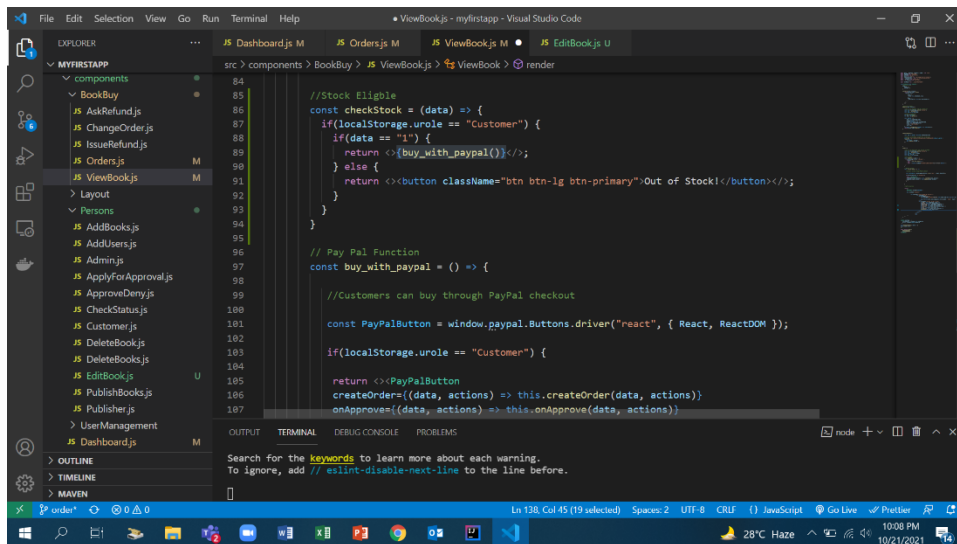
```
1 REACT_APP_LOGIN_BASE_URL = http://loginmicroservices-env.eba-6d9wzpu.ap-southeast-1.elasticbeanstalk.com/
2 REACT_APP_BOOK_BASE_URL = http://bookmicroservices-env.eba-fs3m353v.ap-southeast-1.elasticbeanstalk.com/
3 REACT_APP_ORDER_BASE_URL = http://ordermicroservices-env.eba-9xyghzjp.ap-southeast-1.elasticbeanstalk.com/
4 REACT_APP_REVIEW_BASE_URL = http://reviewmicroservices-env.eba-jzevrwka.ap-southeast-1.elasticbeanstalk.com/
5
6 REACT_APP_LOGIN_BASE_URL_REMOVE = http://localhost:8080/
7 REACT_APP_BOOK_BASE_URL_REMOVE = http://localhost:8081/
8 REACT_APP_ORDER_BASE_URL_REMOVE = http://localhost:8082/
9 REACT_APP_REVIEW_BASE_URL_REMOVE = http://localhost:8083/
10
11
12 REACT_APP_AWS_BUCKET_NAME = boockeroo
13 REACT_APP_AWS_REGION_NAME = ap-southeast-1
14 REACT_APP_AWS_ACCESS_KEY_ID_NAME = AKIAYLAR4JVN266ZONV
15 REACT_APP_AWS_SECRET_ACCESS_KEY_NAME = 1IdLS5kqJMa978eU2AvroKa+tk7Ql67dZyjr7WhP
16
```

With such a coding style, it is easier to switch from production to deployment.

Also brought changes into the PayPal API. Earlier, we used a NPM package. Now we are using the direct API from PayPal. However, this wasn't enough, as before allowing the user to pay, we needed to ensure whether the book was in stock or not. Hence made a function that checks first whether a book is in stock or not.



Over here in the image, I am commenting out the previous function.



The **checkStock()** ensures if a book is in stock or not and then renders the PayPal button to the user. The pretty much were the refactoring done to the website.

Gitflow Organisation

Screenshot of all GitHub branches are shown below to show our Gitflow organisation.

Note: the screenshot of each branch shows how much commits we have done for each branch. Please also note, we kept the frontend/backend developers to do the commits but other members contributed in the meetings/in chat for MS Teams to development/enhancements of code.

Main:

main	7 branches	0 tags	Go to file	Add file	Code
DavidAziz3825605	Update .env	✓ c7f2a0e 29 minutes ago	🕒 98 commits		
.circleci	CircleCI yml added			last month	
.idea	rmidea			last month	
BackEnd	Added a spelling error in the docker file of review			1 hour ago	
Documentation	Uploading/Updating Milestone 2 Submission			last month	
FrontEnd/myfirstapp	Update .env			29 minutes ago	
.gitignore	Initial commit			3 months ago	
README.md	Update README.md			2 months ago	

Approve-deny-user branch:

approve-deny-u...	7 branches	0 tags	Go to file	Add file	Code
This branch is 47 commits ahead of main.			Contribute		
KowsarRahman	Dockerfile updated	✓ bb19c2d on 21 Sep	🕒 48 commits		
.circleci	CircleCI yml added			last month	
.idea	Publisher registration done and API endpoint created			last month	
BackEnd	Dockerfile updated			last month	
Documentation	Uploading/Updating Milestone 2 Submission			last month	
FrontEnd/myfirstapp	Front-End testings added			last month	
.gitignore	Initial commit			3 months ago	
README.md	Update README.md			2 months ago	

Book branch:

book 7 branches 0 tags Go to file Add file Code

This branch is 40 commits ahead of main. Contribute

En1gma64 Unit test bug fix 673c330 on 21 Sep 41 commits

.idea	Publisher registration done and API endpoint created	last month
BackEnd	Unit test bug fix	last month
Documentation	Milestone 2 doc uploaded	last month
FrontEnd/myfirstapp	React App Dockerised and built jar files for Java microservices	last month
.gitignore	Initial commit	3 months ago
README.md	Update README.md	2 months ago

Develop Branch:

develop 7 branches 0 tags Go to file Add file Code

This branch is up to date with main. Contribute

DavidAziz3825605 Update .env c7f2a0e 28 minutes ago 98 commits

.circleci	CircleCI yml added	last month
.idea	rmidea	last month
BackEnd	Added a spelling error in the docker file of review	1 hour ago
Documentation	Uploading/Updating Milestone 2 Submission	last month
FrontEnd/myfirstapp	Update .env	28 minutes ago
.gitignore	Initial commit	3 months ago
README.md	Update README.md	2 months ago

Login-register Branch:


login-register 7 branches 0 tags Go to file Add file Code


This branch is 16 commits ahead of main. Contribute


KowsarRahman Added user based action templates 03b5125 on 16 Sep 17 commits

.idea	Publisher registration done and API endpoint created	last month
BackEnd/loginmicroservices	Publisher registration done and API endpoint created	last month
Documentation	Added wireframes	2 months ago
FrontEnd/myfirstapp	Added user based action templates	last month
.gitignore	Initial commit	3 months ago
README.md	Update README.md	2 months ago

Order Branch:

 order ▾


 7 branches


 0 tags








Go to file

Add file ▾


Code ▾


This branch is 69 commits ahead of main.  Contribute ▾


 **En1gma64** App.properties adjustments ✓ 62a3717 2 days ago ⌚ 70 commits

 .circleci	CircleCI yml added	last month
 .idea	rmidea	last month
 BackEnd	App.properties adjustments	2 days ago
 Documentation	Uploading/Updating Milestone 2 Submission	last month
 FrontEnd/myfirstapp	Publisher/Admin cad edit price and mark out of stock, price bug solved	2 days ago
 .gitignore	Initial commit	3 months ago
 README.md	Update README.md	2 months ago

Review Branch:

 review ▾


 7 branches


 0 tags








Go to file

Add file ▾

Code ▾

This branch is 83 commits ahead of main.  Contribute ▾

 **KowsarRahman** Merge branch 'review' of <https://github.com/DavidAziz3825605/G03-...> ✓ 461bb7d 5 hours ago ⌚ 84 commits

 .circleci	CircleCI yml added	last month
 .idea	rmidea	last month
 BackEnd	Merge branch 'review' of https://github.com/DavidAziz3825605/G03-...	5 hours ago
 Documentation	Uploading/Updating Milestone 2 Submission	last month
 FrontEnd/myfirstapp	Review feature completely finished	17 hours ago
 .gitignore	Initial commit	3 months ago
 README.md	Update README.md	2 months ago

Scrum Process

Now as a group we had each member assigned to particular roles:

Rashed Abdin: Scrum Master/Product Owner

Kowsar Rahman: Front End Developer/Databasing

Dunith Nadvi Karawita: Back End Developer/Testing

David Aziz: Helping everyone in tasks

Please Note: Since we lost a team member (Oliver), we all had to reassign to different roles and take added responsibility.

The way we organized our work and process were through making sure everyone completed their assigned tasks for their specific roles they have agreed to take. But that being said, we all helped each other implement and complete all tasks throughout the entire project. We found it easier to assign roles to each member who are confident in taking up that role. For example, Kowsar is very confident in implementing the front-end, so we agreed as a group for him to take that role. We had our scrum process so that each member works in an area they are confident in rather than each member being all over the place completing tasks in different areas. We all liked that idea and decided to try it for milestone 1 and we find it to work very well. Hence, we kept this scrum process throughout all sprints and managed to complete everything at a very high level for each milestone.

Deployment Pipeline

This screenshot shows the CircleCI interface for the pipeline 'G03-Software.Eng-Tools-and-Process'. The pipeline is currently in a 'Success' state. The workflow 'say-hello-workflow' is shown with a job 'say-hello' that completed successfully 8 seconds ago. The branch/commit is 'review' with commit hash '461bb7d'. The commit message is 'Merge branch 'review' of https://github.com/DavidAziz3825605/G03-Software.Eng-Tools-and-Process into review'.

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
G03-Software.Eng-Tools-and-Process 35	Success	say-hello-workflow	review 461bb7d Merge branch 'review' of https://github.com/DavidAziz3825605/G03-Software.Eng-Tools-and-Process into review	6h ago	13s	...
Jobs						
	Success	say-hello			8s	

This screenshot shows the CircleCI interface for the pipeline 'G03-Software.Eng-Tools-and-Process'. The pipeline is currently in a 'Success' state. The workflow 'say-hello-workflow' is shown with a job 'say-hello' that completed successfully 3 seconds ago. The branch/commit is 'develop' with commit hash '2eba236'. The commit message is 'Added a spelling error in the docker file of review'.

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
G03-Software.Eng-Tools-and-Process 45	Success	say-hello-workflow	develop 2eba236 Added a spelling error in the docker file of review	6m ago	7s	...
Jobs						
	Success	say-hello			3s	

Basically, the job of CircleCi is to make a docker image every time a new commit is pushed. This increases automation and ensures continuous deployment.

Log4J Screenshots:

```
2021-10-23 14:42:12.624 TRACE 17952 --- [nio-8082-exec-2] c.n.s.o.Web.OrderController : Retrieved orders by seller: Kaif Rahman
2021-10-23 14:42:12.636 TRACE 17952 --- [nio-8082-exec-10] c.n.s.o.Web.OrderController : Retrieved orders by user: kaif@kaif.com
2021-10-23 14:42:12.638 TRACE 17952 --- [nio-8082-exec-4] c.n.s.o.Web.OrderController : Retrieved orders that are available for ref
2021-10-23 14:42:12.666 TRACE 17952 --- [nio-8082-exec-3] c.n.s.o.Web.OrderController : Retrieved all orders
2021-10-23 14:42:13.708 TRACE 17952 --- [nio-8082-exec-5] c.n.s.o.Web.OrderController : Updated order
2021-10-23 14:42:13.840 TRACE 17952 --- [nio-8082-exec-10] c.n.s.o.Web.OrderController : Retrieved orders by seller: Kaif Rahman
2021-10-23 14:42:13.846 TRACE 17952 --- [nio-8082-exec-9] c.n.s.o.Web.OrderController : Retrieved orders by user: kaif@kaif.com
2021-10-23 14:42:13.846 TRACE 17952 --- [nio-8082-exec-2] c.n.s.o.Web.OrderController : Retrieved orders that are available for ref
2021-10-23 14:42:13.857 TRACE 17952 --- [nio-8082-exec-4] c.n.s.o.Web.OrderController : Retrieved all orders
2021-10-23 14:42:14.072 TRACE 17952 --- [nio-8082-exec-10] c.n.s.o.Web.OrderController : Retrieved orders by seller: Admin
2021-10-23 14:42:14.074 TRACE 17952 --- [nio-8082-exec-6] c.n.s.o.Web.OrderController : Retrieved orders by user: admin@admin.com
2021-10-23 14:42:14.078 TRACE 17952 --- [nio-8082-exec-5] c.n.s.o.Web.OrderController : Retrieved orders that are available for ref
2021-10-23 14:42:14.088 TRACE 17952 --- [nio-8082-exec-9] c.n.s.o.Web.OrderController : Retrieved all orders
2021-10-23 14:42:14.370 TRACE 17952 --- [nio-8082-exec-6] c.n.s.o.Web.OrderController : Retrieved orders by seller: Rokomari Books
2021-10-23 14:42:14.373 TRACE 17952 --- [nio-8082-exec-1] c.n.s.o.Web.OrderController : Retrieved orders by user: rokoman@gmail.co
2021-10-23 14:42:14.376 TRACE 17952 --- [nio-8082-exec-10] c.n.s.o.Web.OrderController : Retrieved orders that are available for ref
2021-10-23 14:42:14.385 TRACE 17952 --- [nio-8082-exec-5] c.n.s.o.Web.OrderController : Retrieved all orders
2021-10-23 14:43:05.381 TRACE 17952 --- [nio-8082-exec-5] c.n.s.o.Web.OrderController : Updated order
2021-10-23 14:43:05.423 TRACE 17952 --- [nio-8082-exec-7] c.n.s.o.Web.OrderController : Retrieved orders by seller: Rokomari Books
2021-10-23 14:43:05.427 TRACE 17952 --- [nio-8082-exec-9] c.n.s.o.Web.OrderController : Retrieved orders by user: rokoman@gmail.co
2021-10-23 14:43:05.431 TRACE 17952 --- [nio-8082-exec-3] c.n.s.o.Web.OrderController : Retrieved orders that are available for ref
2021-10-23 14:43:05.434 TRACE 17952 --- [nio-8082-exec-4] c.n.s.o.Web.OrderController : Retrieved all orders
```

```
2021-10-23 14:24:22.591 INFO 8264 --- [nio-8083-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatc
2021-10-23 14:24:22.594 INFO 8264 --- [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2021-10-23 14:24:22.642 INFO 8264 --- [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 48 ms
2021-10-23 14:24:23.019 TRACE 8264 --- [nio-8083-exec-2] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 1234987654
2021-10-23 14:26:22.099 TRACE 8264 --- [nio-8083-exec-4] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 1234987654
2021-10-23 14:26:26.527 TRACE 8264 --- [nio-8083-exec-5] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 1234987654
2021-10-23 14:26:43.780 TRACE 8264 --- [nio-8083-exec-8] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 6978123411567
2021-10-23 14:27:23.668 TRACE 8264 --- [nio-8083-exec-10] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 1234987654
2021-10-23 14:27:49.882 TRACE 8264 --- [nio-8083-exec-2] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 1234987654
2021-10-23 14:27:57.603 TRACE 8264 --- [nio-8083-exec-3] c.n.s.r.Web.ReviewController : Deleted review with ID: 15
2021-10-23 14:27:57.603 TRACE 8264 --- [nio-8083-exec-4] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 1234987654
2021-10-23 14:28:26.850 TRACE 8264 --- [nio-8083-exec-6] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 6978123411567
2021-10-23 14:28:44.887 TRACE 8264 --- [nio-8083-exec-8] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 1234987654
2021-10-23 14:28:56.394 TRACE 8264 --- [nio-8083-exec-1] c.n.s.r.Web.ReviewController : Created new review
2021-10-23 14:28:56.783 TRACE 8264 --- [nio-8083-exec-2] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 1234987654
2021-10-23 14:29:02.233 TRACE 8264 --- [nio-8083-exec-4] c.n.s.r.Web.ReviewController : Deleted review with ID: 16
2021-10-23 14:29:16.613 TRACE 8264 --- [nio-8083-exec-6] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 1234987654
2021-10-23 14:30:51.222 TRACE 8264 --- [nio-8083-exec-8] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 1234987654
2021-10-23 14:31:09.278 TRACE 8264 --- [nio-8083-exec-10] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 1234987654
2021-10-23 14:35:48.305 TRACE 8264 --- [nio-8083-exec-2] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 123422998811
2021-10-23 14:35:52.063 TRACE 8264 --- [nio-8083-exec-4] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 8948398489148
2021-10-23 14:41:57.273 TRACE 8264 --- [nio-8083-exec-7] c.n.s.r.Web.ReviewController : Retrieved all reviews with ISBN: 3334343434343
```

This is a tool called Log4J which helps track things back if something goes wrong by logging the code.

Docker:

We used docker for deployment

Login Microservices:

```
FROM openjdk:8
EXPOSE 8080
ADD target/loginmicroservices-docker.jar loginmicroservices-docker.jar
ENTRYPOINT ["java", "-jar", "/loginmicroservices-docker.jar"]
```

Front-End DockerFile:

```
1 FROM node:14
2 WORKDIR /app
3 COPY package*.json ./
4 RUN npm install
5 COPY . .
6 EXPOSE 8080
7 CMD ["npm", "start"]
```

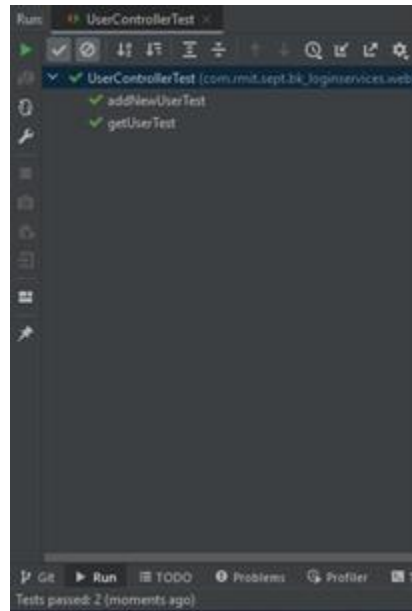
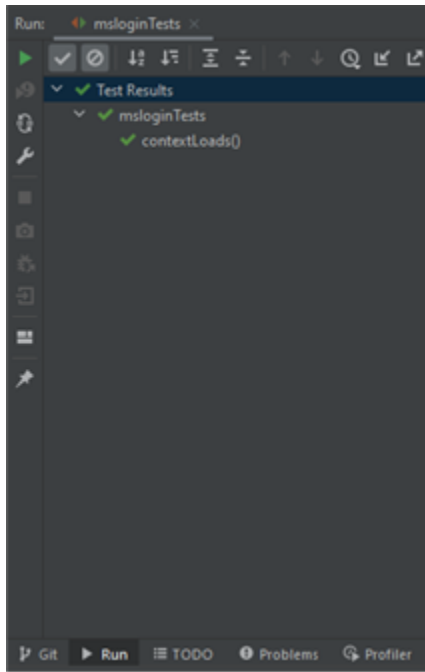
Book Microservices:

```
Dockerfile x
1 FROM openjdk:8
2 EXPOSE 8081
3 ADD target/bookmicroservices-0.0.1-SNAPSHOT.jar bookmicroservices-0.0.1-SNAPSHOT.jar
4 ENTRYPOINT ["java", "-jar", "/bookmicroservices-0.0.1-SNAPSHOT.jar"]
```

Test Documentation/Execution

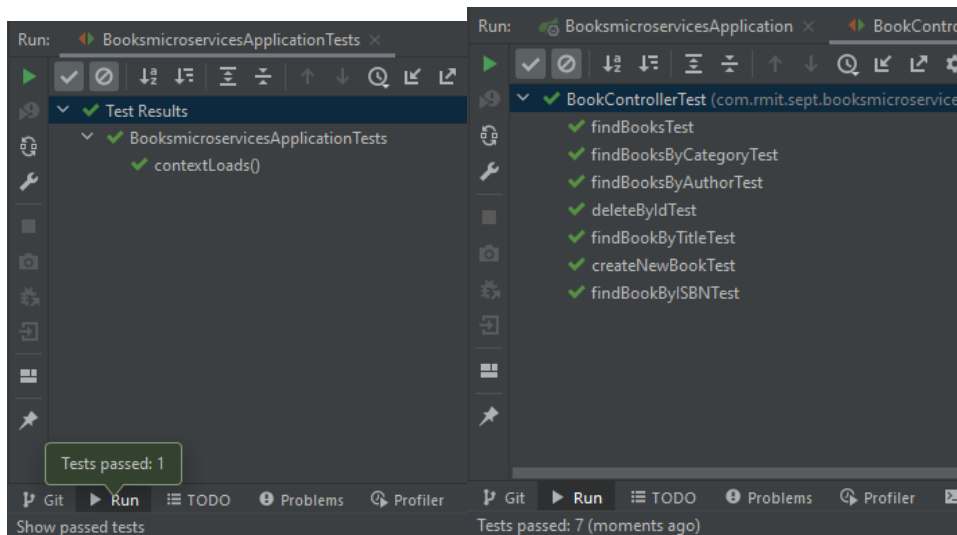
Unit tests for login microservice:

Tested adding a new user and getting a user from a mock repository



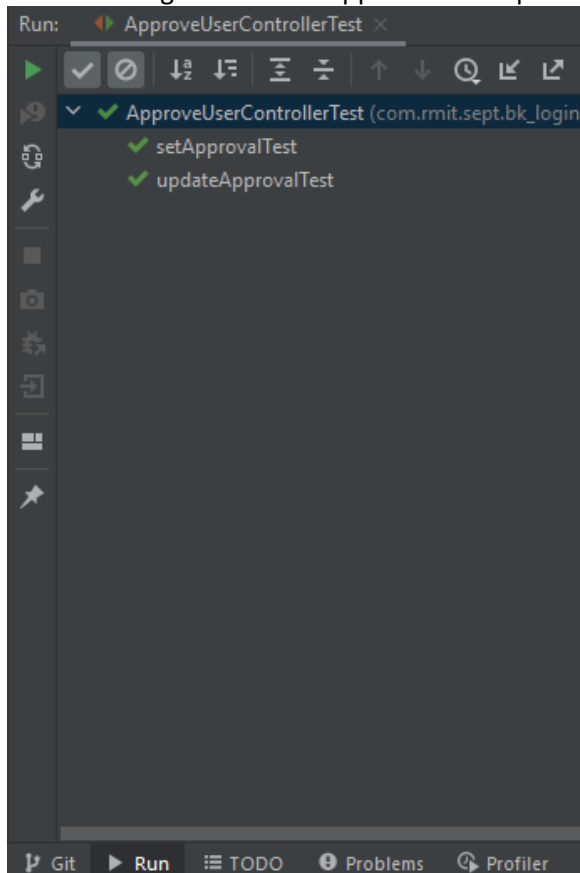
Unit tests for book microservice:

Tested finding books with multiple criteria along with adding, deleting, and updating a book created in a mock repository



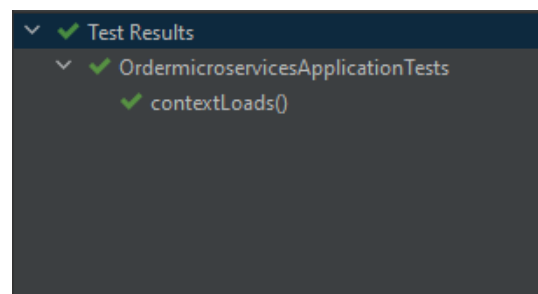
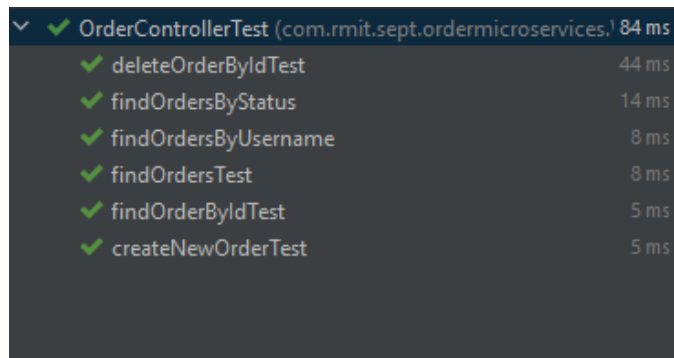
Approve User tests:

Tested adding a user to be approved and updating his/her status in a mock repository



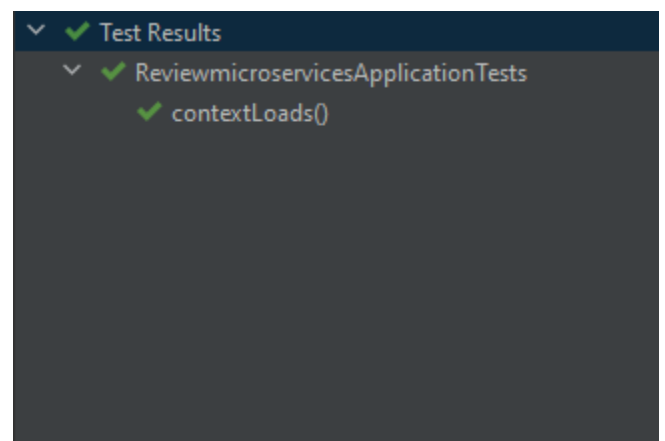
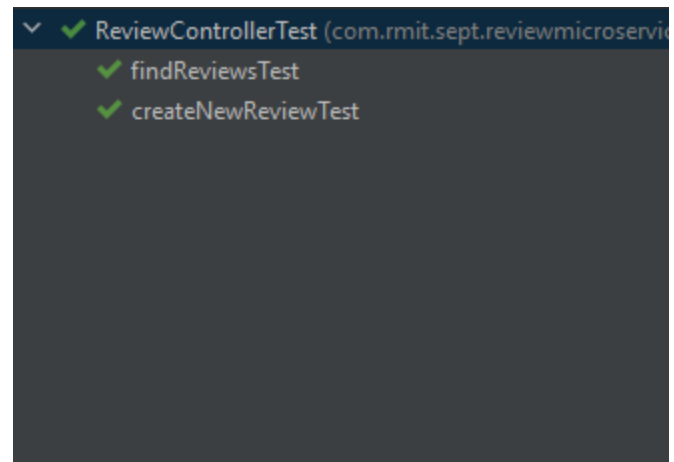
Order Tests:

Testing finding orders with multiple criteria, along with adding, deleting and updating orders in a mock repository



Review Tests:

Testing finding and creating reviews in a mock repository



POSTMAN:

GET http://localhost:8082/api/order/findOrdersByUsernameAndStatus/d.n.karawita@gmail.com/Not Shipped

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 12 ms Size: 440 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 5,
3   "title": "Da Vinci Code",
4   "username": "d.n.karawita@gmail.com",
5   "seller": "Harper Collins",
6   "status": "Not Shipped",
7   "price": 41.0,
8   "createAt": "2021-47-23",
9   "updateAt": null,
10  "isbn": "BN330"
11 }
12
13
```

GET http://localhost:8082/api/order/findOrdersByUsername/d.n.karawita@gmail.com ...

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (9) Test Results

Status: 200 OK Time: 10 ms Size: 866 B

Pretty Raw Preview Visualize JSON

```
15 {
16   "title": "Hunger Games",
17   "username": "d.n.karawita@gmail.com",
18   "seller": "Penguin",
19   "status": "Shipped",
20   "price": 43.0,
21   "createAt": "2021-18-22",
22   "updateAt": null,
23   "isbn": "16546545465464N"
24 },
25 {
26   "id": 1,
27   "title": "Game of thrones",
28   "username": "d.n.karawita@gmail.com",
29   "seller": "Penguin",
30   "status": "Order Placed",
31   "price": 39.0,
32   "createAt": "2021-31-22",
33   "updateAt": null,
34   "isbn": "BN330"
35 }
36 ]
37
```

Activate Window
Go to Settings to activate

GET http://localhost:8082/api/order/findOrdersForRefund/d.n.karawita@gmail.com ...

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (9) Test Results Status: 200 OK Time: 14 ms

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 2,
3   "title": "Hunger Games",
4   "username": "d.n.karawita@gmail.com",
5   "seller": "Penguin",
6   "status": "Shipped",
7   "price": 43.0,
8   "createAt": "2021-10-24",
9   "updateAt": null,
10  "isbn": "16546545465465464N"
11 }
12
13
```

Please note: we have the documentation of acceptance testing in a separate pdf file called Acceptance Testing in this folder where you will find all acceptance tests for each user story 😊