Sports Shoes E-com Applicaication:

Idea:

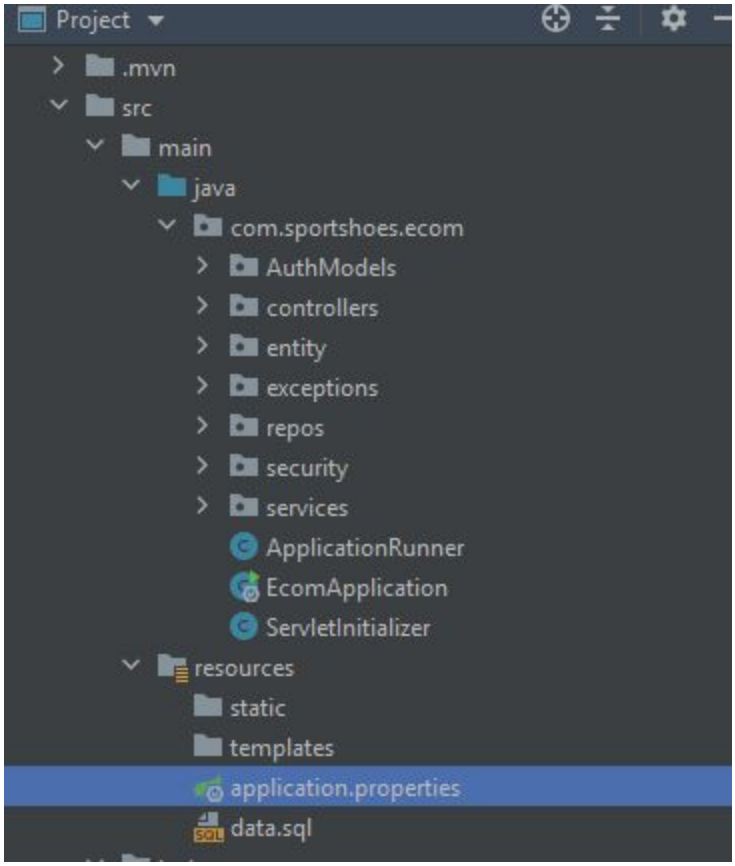
Sports shoes Ecom App provides a set of REST APIs to manage products, categories and customers for an administration and provides feature manage Cart and purchases for an user. Application leverages latest Spring boot configurations/features for efficiency.

TO run the application:

1. Clone the project using the below URL : **https://github.com/sandeshMS1996/SportsShoesEcomApp.git**

2. In the application.properties make sure that the property **spring.profile.active** is not set to **test.**
3. Run the application

Directory structure:



Technologies used:

1. API Documentation: Swagger/Swagger UI.
2. Jenkins for Continuous Integration and continuous Delivery
3. Hibernate JPA
4. Spring boot
5. Spring security

Important features:

1. **Security:** Spring boot security is used to authenticate and authorize users based on his ROLE. currently there are 2 ROLEs defined for the application:

   ROLE_ADMIN: Admin can Add/Remove a product or category and view purchase summary filtered based on Purchase date or the category. He would be able to view all registered users and view his purchases.

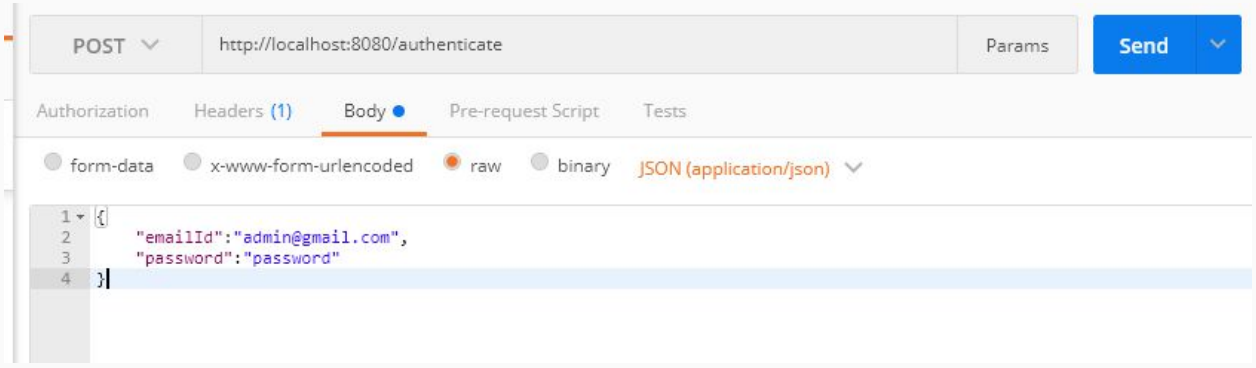
   ROLE_USER: This role is meant for any user who wishes to make purchase by admin/removing a product to his cart and puy products present in the cart.

AUthentication is performed using JsonWebTokens which must be attached to the header of each request which needs authentication and authorisation. For initial authentication, the user must provide his email ID and password as shown below.

URL:

Registration: any user can register first before logging in. his details would be stored into the database upon registration and an appropriate message would be displayed. The password would be hashed with **BCryptPAsswordEncoder** before it is stored.

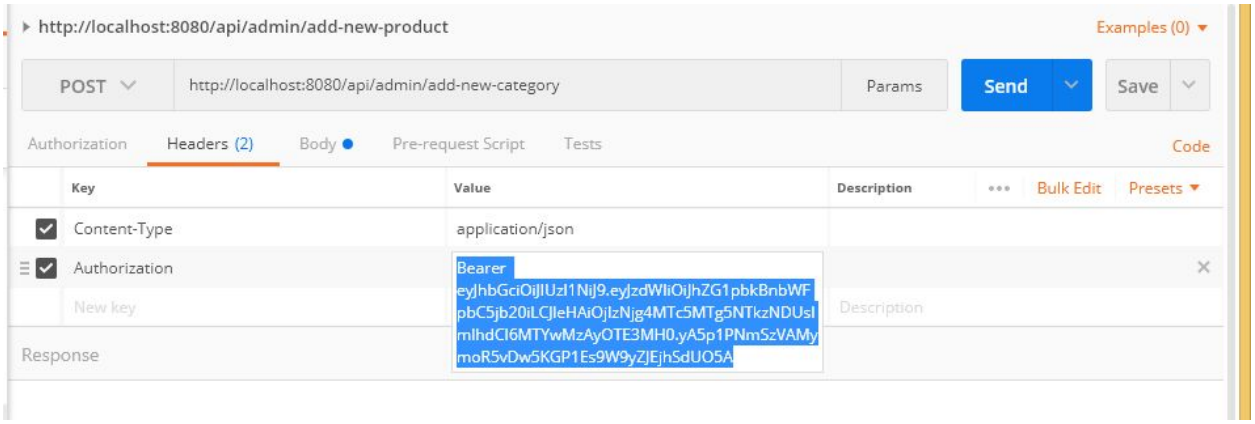
URL: http://localhost:8080/authenticate method : Post:



2. **Admin features:**

To use the below URLs generate JWT token for an admin by authenticating as shown above and add the jwt token as below

Headers: key Authorization , Value: Bearer <JWT String>
Key: Content-Type, value : application/json



An admin would be able to add/remove a product or category and view purchase summary.

Below is the list of URLs assigned only for ADMINs.

URL	Method
/api/admin/add-new-product	POST
/api/admin/add-new-category	POST
/api/admin/get-all-registered-users	GET
/api/admin/purchase-summary	GET
/api/admin/category/delete-category/{id}	DELETE
/api/admin/category/delete-product/{id}	DELETE
/api/admin/account/delete-account/{id}	DELETE
/api/admin/purchase/filter-purchase-by-category/{category ID}	GET
/api/admin/purchase/filter-purchase-by-date/{date in format(yyyy-MM-dd)}	GET

3. **Customer Features:**

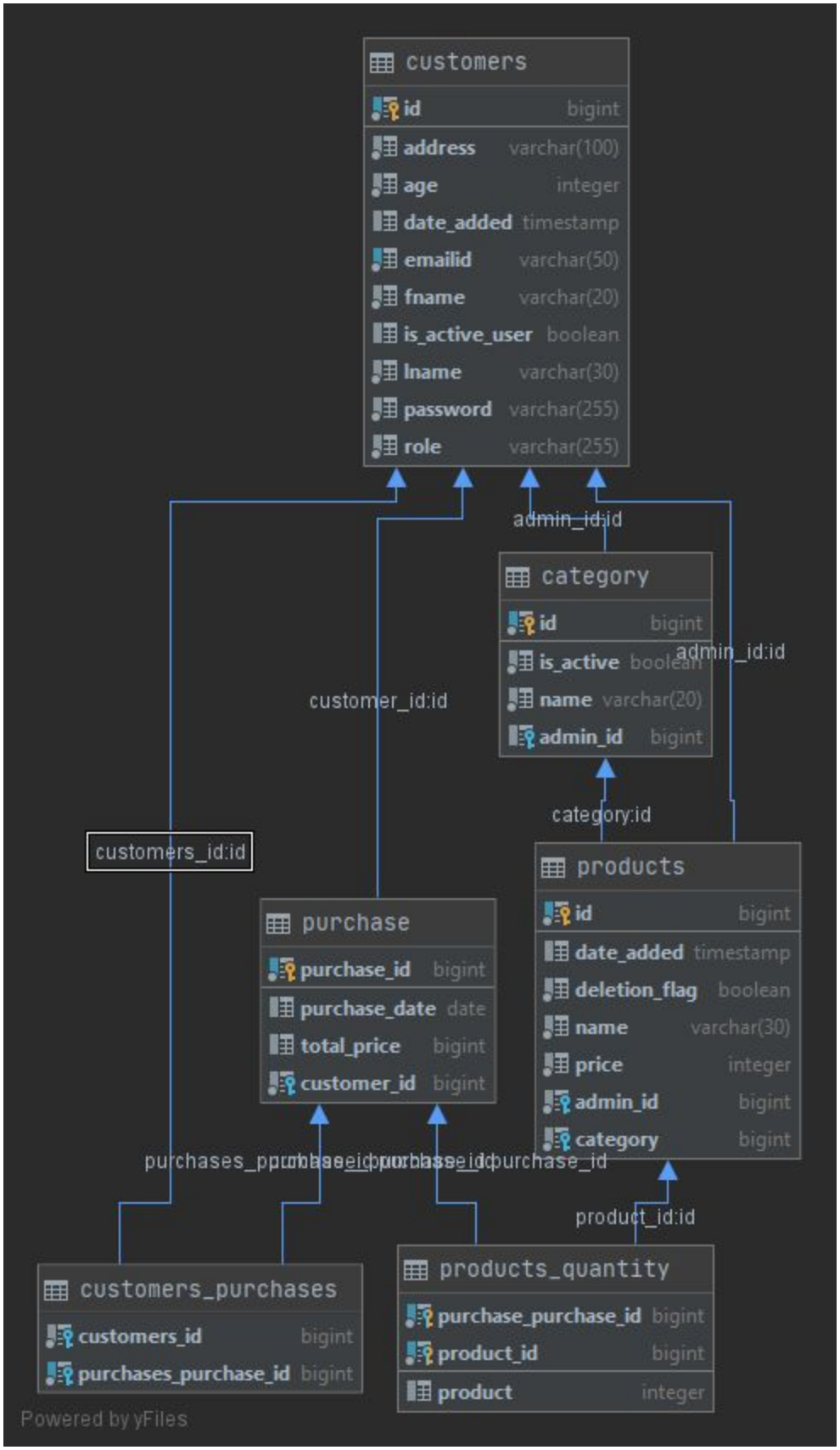
Customers can add a product to cart and view/Buy items in the cart. To perform any of these user has to login with ADMIN/USER role and provide the JWT in the header as shown above:

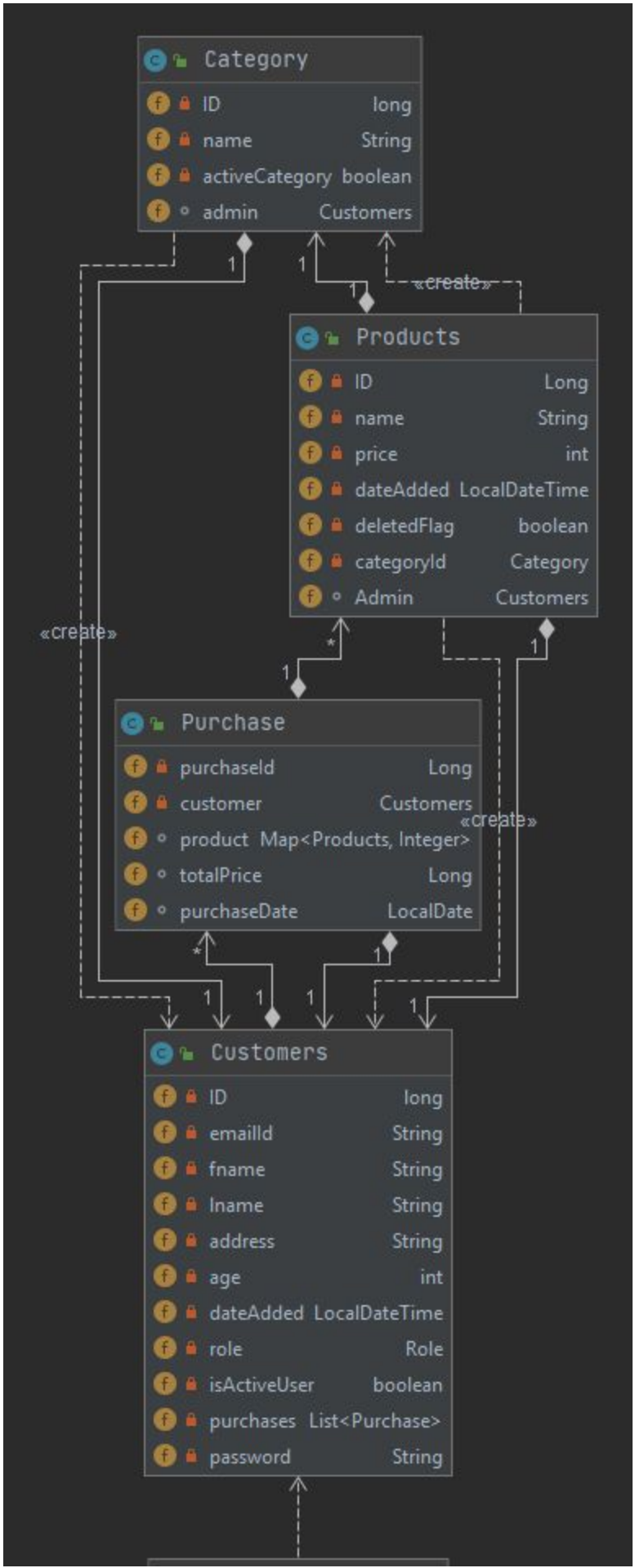
URL	Method
/api/customer/products/add-to-cart	POST
/api/customer/products/view-cart	GET
/api/customer/products/buy-product	POST
/api/customer/products/purchase-history	GET (This method returns the purchases of currently logged in user)
/api/customer/change/password	POST (changes password for currently logged in user)

APIs configured for public user which do not require any authentication:

URL	Method
/api/product/{product id}	GET (returns product details for product id)
/api/products	GET (returns all product)
/api/products/search-by-category/{category ID}	GET
/api/register	POST (USER registration)
/authenticate	POST (used for login)

Schema diagram of the database:


customers
id bigint
address varchar(100)
age integer
date_added timestamp
emailid varchar(50)
fname varchar(20)
is_active_user boolean
lname varchar(30)
password varchar(255)
role varchar(255)
admin_id:id
category
id bigint
is_active boolean
name varchar(20)
admin_id bigint
admin_id:id
customer_id:id
category:id
customers_id:id
products
id bigint
date_added timestamp
deletion_flag boolean
name varchar(30)
price integer
admin_id bigint
category bigint
purchase
purchase_id bigint
purchase_date date
total_price bigint
customer_id bigint
purchase_id
product_id:id
customers_purchases
customers_id bigint
purchases_purchase_id bigint
products_quantity
purchase_purchase_id bigint
product_id bigint
product integer
Powered by yFiles


Category
ID long
name String
activeCategory boolean
admin Customers
«create»
Products
ID Long
name String
price int
dateAdded LocalDateTime
deletedFlag boolean
categoryId Category
Admin Customers
«create»
Purchase
purchaseId Long
customer Customers
product Map<Products, Integer>
totalPrice Long
purchaseDate LocalDate
«create»
Customers
ID long
emailId String
fname String
lname String
address String
age int
dateAdded LocalDateTime
role Role
isActiveUser boolean
purchases List<Purchase>
password String

Software development details:

Software Development Model: AJILE
Number of Sprints : 3
Duration of each Sprint : 1 Week(5 Days)

Details of each Sprint:

Sprint 1:

User story 1:

As a user/Admin, i need to register and login to the application so that i can access the app

Description: accept personal details from h user and and store it into a database, make sure Email ID is unique for each user.

Accept Email ID and password form the registered user and validate the]m against the details stored in the database, provide access only if successfully validated.

Acceptance criteria:

1. All details must be validated before persisting them and display appropriate message.
2. Email id should be unique to each other, reject any attempts to register with duplicate email ID

User Story 2:

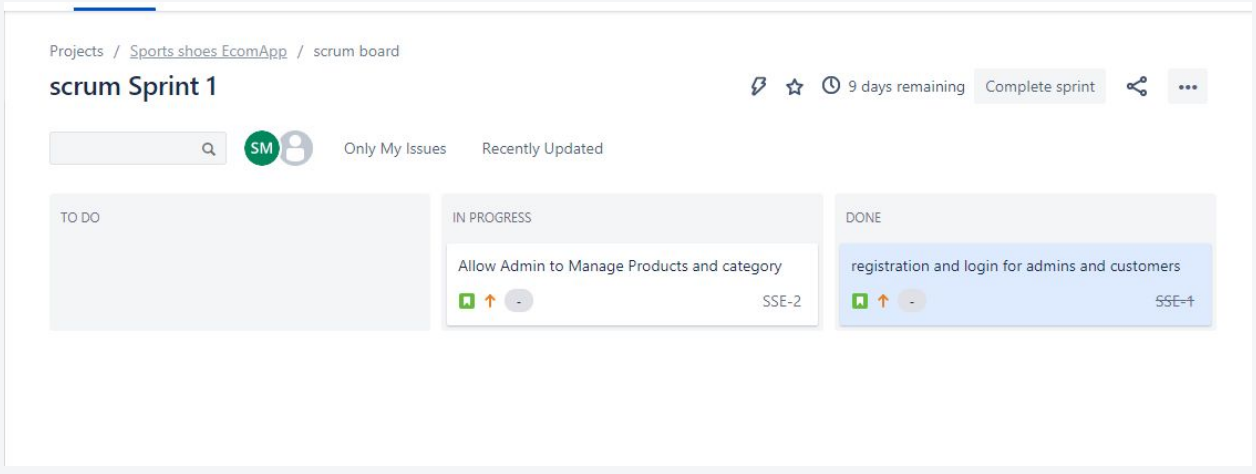
As a Admin of the application, I need to add, delete or modify Products and categories as per the need so that i can manage the application efficiently.

Description: Create APIs to create, delete and modify products and categories and store them into the database. upon successful persistence, display appropriate message .

Acceptance criteria:

1. Only Admins should be able to access these features and the effects should be cascading i.e. if a category is deleted, all products associated with that category should be deleted.

Jira screenshot for Sprin1



Sprint 2:

USer story 1:

Ad a customer, I need to view all available products, filter them based on the criteria

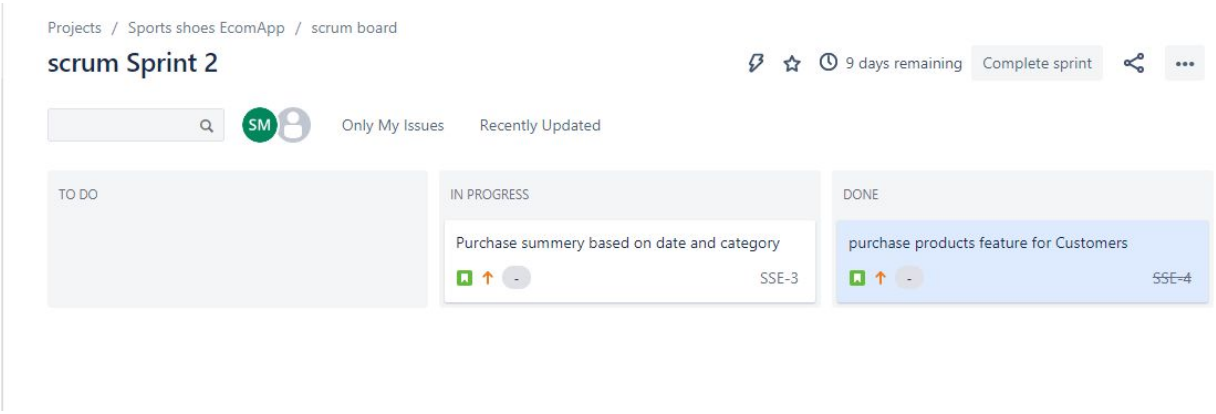
Description: Create API to allow customer view Products and filter them based on the category, alos allow them to view individual products based on selection

USer story 2:

As an admin , I need to view Purchase summary and should be able to categories them based on Category and purchase date.

Description: create API to View all purchases and filter them based on Category/Purchase date.Create API to fetch the details of all registered users

Success criteria: Only admin should be able to access these features.

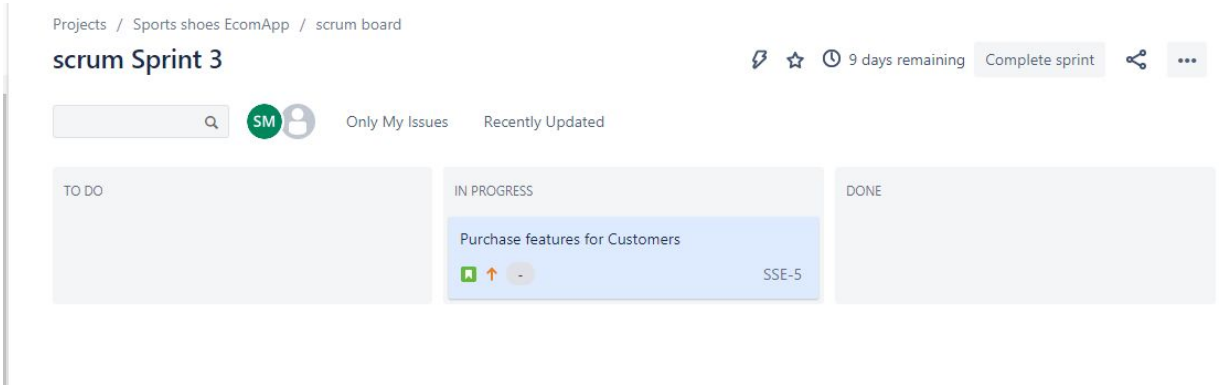


Sprint 3:

User story 1:

As a customer, I need to manage my carts by adding/deleting the items to cart and buying them at once.

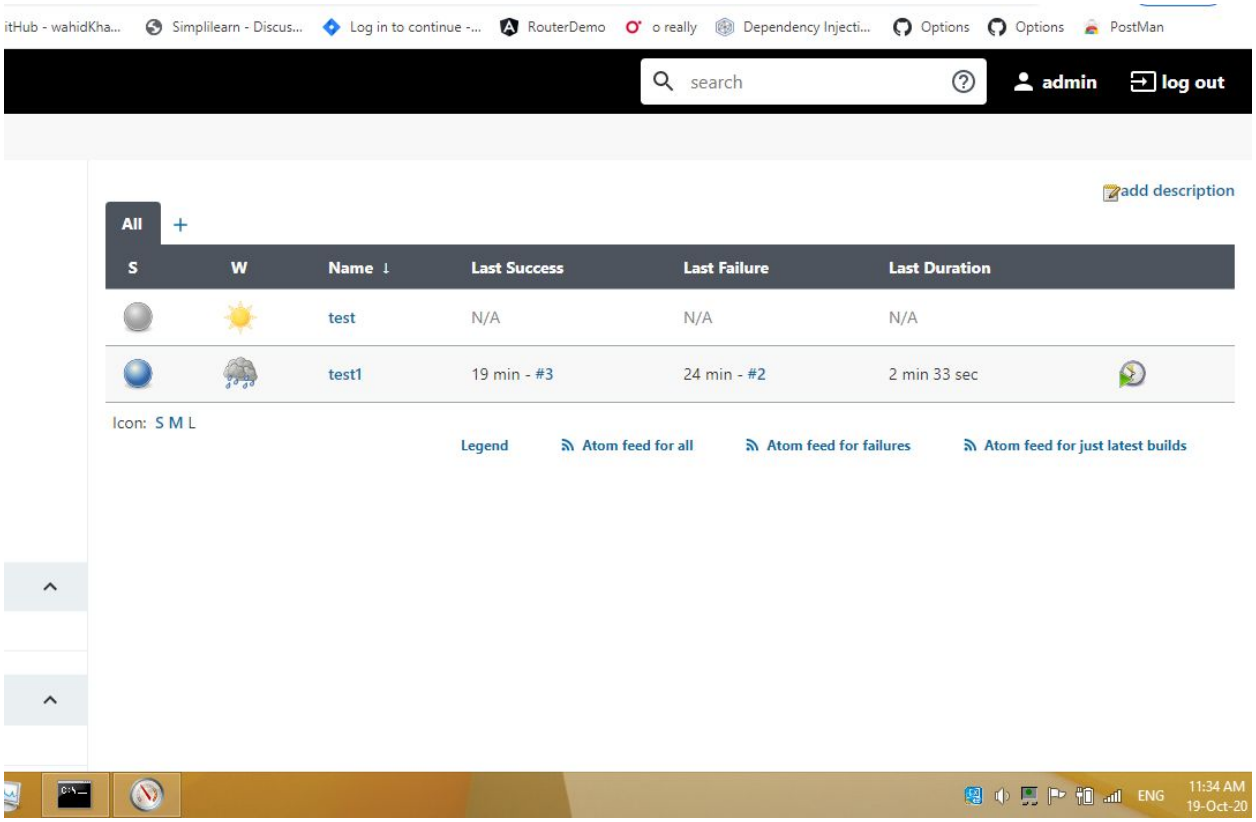
Description: Create APIs to add items to cart and view the available items in the cart and buy them.



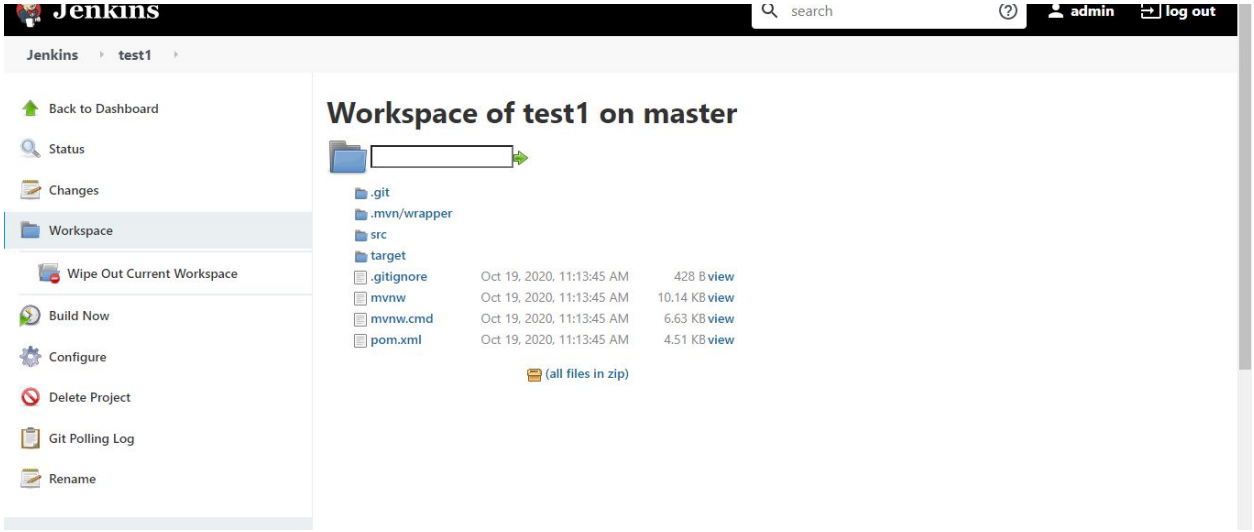
Continuous Integration and continuous delivery :

Jenkins Screenshots:

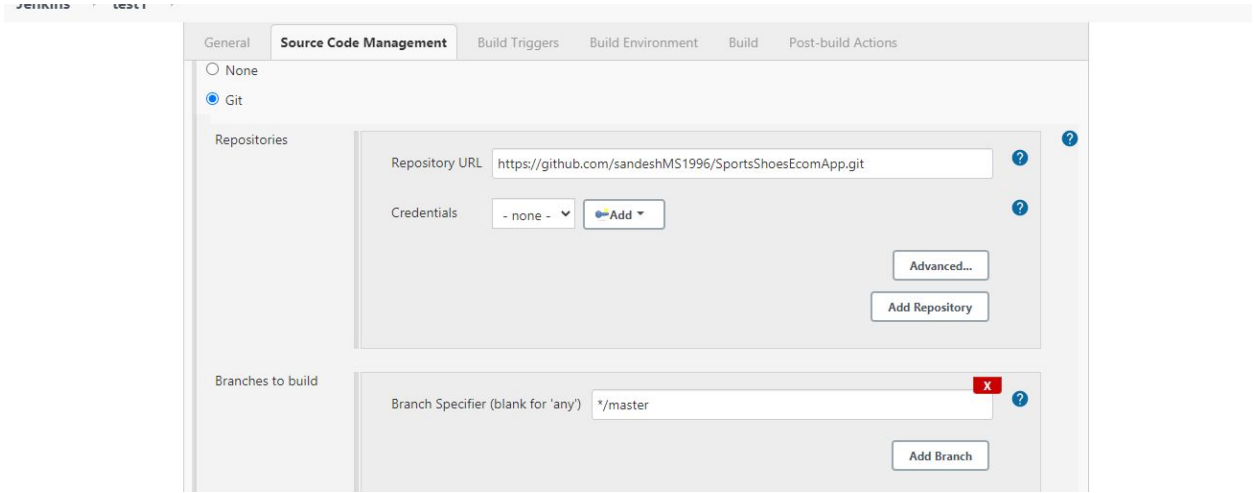
Jenkins dashboard:

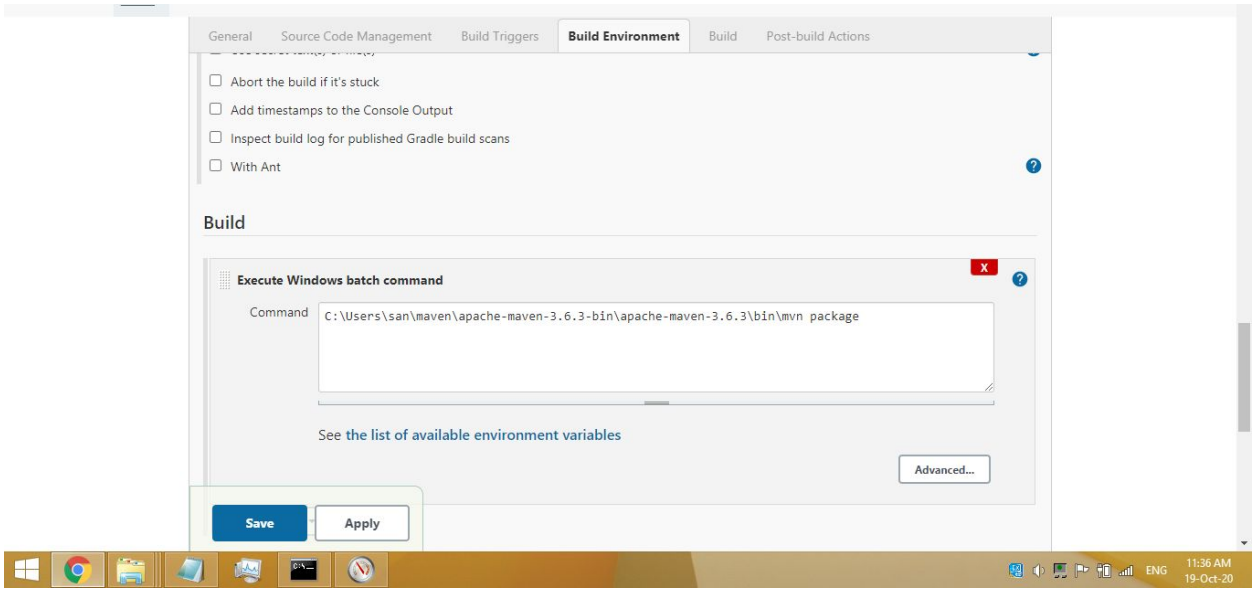


Jenkins workspace:

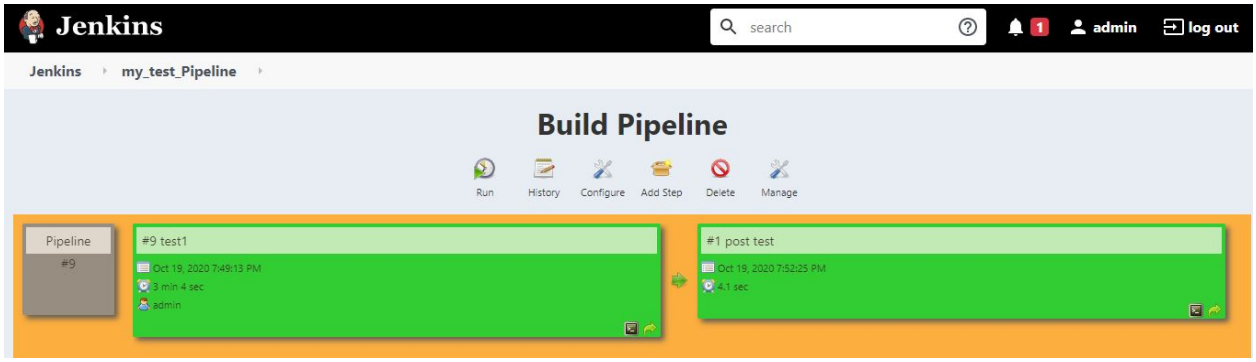

Jenkins
search
admin
log out
Jenkins
test1
Back to Dashboard
Status
Changes
Workspace
Wipe Out Current Workspace
Build Now
Configure
Delete Project
Git Polling Log
Rename
Workspace of test1 on master
.git
.mvn/wrapper
src
target
.gitignore Oct 19, 2020, 11:13:45 AM 428 B view
mvnw Oct 19, 2020, 11:13:45 AM 10.14 KB view
mvnw.cmd Oct 19, 2020, 11:13:45 AM 6.63 KB view
pom.xml Oct 19, 2020, 11:13:45 AM 4.51 KB view
(all files in zip)

Workspace:


General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
None
Git
Repositories
Repository URL
https://github.com/sandeshMS1996/SportsShoesEcomApp.git
Credentials
- none -
Add
Advanced...
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/master
Add Branch



Build PipeLine for Demonstration:(POST test has no significance here, it just runs a echo command to demonstrate the pipeline)



Conclusion:

OverAll it is a good application starting point to bring E Commerce features for sprotes shoes store

Below are some of the some of Enhancement that would be applied in future.

1 Allow admins to add images for each products to improve the presentation of the products to the customer.

2. Build a front end that leverages these APis to present users products to the user and buying them.