DOCUMENTATION GROUP NO: 4

PROJECT NAME: PHARMACY

Milestone 1: DATA SERVICE

STORIES:

Story 1:-

Save method for User

<S extends T>S save (S User)

This method is used to insert data. Here we are using save method to add a new user to a context (instance of database context) which will insert a new record in database or saves a given user.

Parameters:

user - Must not be null.

Returns:

Saved user will never be null.

TIME: 40 mins

Story 2:

Save method for Haircare Product.

<S extends T>S save (S Haircare)

This method is used to insert data. Here we are using save method to add a new Haircare product to a context (instance of database context) which will insert a new record in database or saves a given Haircare product.

Parameters:

Haircare - Must not be null.

Returns:

Saved Haircare will never be null.

TIME: 40 mins

Story 3:

Save method for Bodycare Product.

<S extends T>S save (S Bodycare)

This method is used to insert data. Here we are using save method to add a new Bodycare product to a context (instance of database context) which will insert a new record in database or saves a given Bodycare product.

Parameters:

Bodycare - Must not be null.

Returns:

Saved Bodycare will never be null.

TIME: 40 mins

Story 4:

Save method for Skincare Product.

<S extends T>S save (S SkinCare)

This method is used to insert data. Here we are using save method to add a new SkinCare product to a context (instance of database context) which will insert a new record in database or saves a given SkinCare.

Parameters:

SkinCare - Must not be null.

Returns:

Saved SkinCare will never be null.

TIME: 40 mins

Story 5:

Save method for Medicine.

<S extends T>S save (S Medicine)

This method is used to insert data. Here we are using save method to add a new Medicine to a context (instance of database context) which will insert a new record in database or saves a given Medicine.

Parameters:

Medicine - Must not be null.

Returns:

Saved Medicine will never be null.

TIME: 40mins

Story 6:

Save method for Babycare Product.

<S extends T>S save (S BabyCare)

This method is used to insert data. Here we are using save method to add a new babycare product to a context (instance of database context) which will insert a new record in database or saves a given BabyCare.

Parameters:

BabyCare - Must not be null.

Returns:

Saved BabyCare will never be null.

TIME:40 mins

Story 7:

SaveAll method for User.

<S extends T> Iterable <S> saveAll(Iterable<S> User)

This method is used to save all given User into the database. And it also extends iterable.

Parameter

User - Must not be null.

Returns:

Saved User will never be null.

Throws:

IllegalArgumentException- Incase if the given User is null.

TIME: 40mins

Story 8:

SaveAll method for Bodycare Product

<S extends T> Iterable <S> saveAll(Iterable<S> Bodycare)

This method is used to save all given Bodycare into the database. And it also extends iterable.

Parameter

Bodycare :- Must not be null.

Returns:

Saved Bodycare will never be null.

Throws:

IllegalArgumentException: Incase if the given Bodycare is null.

TIME: 40 mins

Story 9:

SaveAll method for Haircare Product.

<S extends T> Iterable <S> saveAll(Iterable<S> Haircare)

This method is used to save all given Haircare into the database. And it also extends iterable.

Parameter

Haircare :- Must not be null.

Returns:

Saved Haircare will never be null.

Throws:

IllegalArgumentException: Incase if the given Haircare is null

TIME: 40 mins

Story 10:

SaveAll method for Skincare Product

<S extends T> Iterable <S> saveAll(Iterable<S> Skincare)

This method is used to save all given Skincare into the database. And it also extends iterable.

Parameter

Skincare :- Must not be null.

Returns:

Saved Skincare will never be null.

Throws:

IllegalArgumentException: Incase if the given Skincare is null.

TIME: 40 mins

Story 11:

SaveAll method for Medicine.

<S extends T> Iterable <S> saveAll(Iterable<S> Medicine)

This method is used to save all given Medicine into the database. And it also extends iterable.

Parameters:

Medicine :- Must not be null.

Returns:

Saved Medicine will never be null.

Throws:

IllegalArgumentException: Incase if the given Medicine is null.

TIME: 40mins

Story 12:

SaveAll method for Babycare Product.

<S extends T> Iterable <S> saveAll(Iterable<S> Babycare)

This method is used to save all given Babycare into the database. And it also extends iterable.

Parameters

Babycare :- Must not be null.

Returns:

Saved Babycare will never be null.

Throws:

IllegalArgumentException: Incase if the given Babycare is null.

TIME: 40mins

Story 13:

findById method for User

[Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) Userid)

This method is used to retrieve or fetch a particular entity by its given Userid.

**Parameters:**

Userid - must not be null.

**Returns:**

the entity with the given Userid or Optional#empty() if none found

**Throws:**

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if Userid is null.

TIME: 40mins

Story 14:

findById method for Bodycare Product

[Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) BodycareId)

This method is used to retrieve or fetch a particular entity by its given BodycareId.

**Parameters:**

BodycareId - must not be null.

**Returns:**

The entity with the given BodycareId or Optional#empty() if none found

**Throws:**

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if BodycareId is null.

TIME: 40mins

Story 15:

findById method for Haircare Product

[Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) HaircareId)

This method is used to retrieve or fetch a particular entity by its given HaircareId.

**Parameters:**

HaircareId - must not be null.

**Returns:**

The entity with the given HaircareId or Optional#empty() if none found

**Throws:**

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if HaircareId is null.

TIME: 40mins

Story 16:

findById method for Skincare product

[Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) SkincareId)

This method is used to retrieve or fetch a particular entity by its given SkincareId.

**Parameters:**

SkincareId - must not be null.

**Returns:**

The entity with the given SkincareId or Optional#empty() if none found

**Throws:**

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if SkincareId is null.

TIME: 40mins

Story 17:

findById method for medicine

[Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) MedicineId)

This method is used to retrieve or fetch a particular entity by its given MedicineId.

**Parameters:**

MedicineId - must not be null.

**Returns:**

The entity with the given MedicineId or Optional#empty() if none found

**Throws:**

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if MedicineId is null.

Story 18:

findById method for Babycare Product:

[Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) BabycareId)

This method is used to retrieve or fetch a particular entity by its given BabycareId.

**Parameters:**

BabycareId - must not be null.

**Returns:**

The entity with the given BabycareId or Optional#empty() if none found

**Throws:**

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if BabycareId is null.

Story 19:

findAllById method for User:

[Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findAllById([Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> UserIds)

This method returns all instances of the type with the given UserIds.

Parameters:

UserIds

Returns:

All types of UserIds.

TIME: 40 mins

Story 20:

findAllById method for Babycare Product:

[Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findAllById([Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> BabycareIds)

This method returns all instances of the type with the given BabycareIds.

Parameters:

BabycareIds

Returns:

All types of BabycareIds.

TIME: 40mins

Story 21:

findAllById method for Haircare Product

[Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findAllById([Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> HaircareIds)

This method returns all instances of the type with the given HaircareIds.

Parameters:

HaircareIds

Returns:

All types of HaircareIds.

TIME: 40mins

Story 22:

findAllById method for Skincare product

[Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findAllById([Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> SkincareIds)

This method returns all instances of the type with the given SkincareIds.

Parameters:

SkincareIds

Returns:

All types of SkincareIds.

TIME: 40mins

Story 23:

findAllById method for medicine

[Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findAllById([Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> MedicineIds)

This method returns all instances of the type with the given MedicineIds.

Parameters:

MedicineIds

Returns:

All types of MedicineIds.

TIME: 40mins

Story 24:

findAllById method for Babycare Products:

[Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findAllById([Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> BabyCareIds)

This method returns all instances of the type with the given BabyCareIds.

Parameters:

BabyCareIds.

Returns:

All types of BabyCareIds.

TIME: 40mins

Story 25:

findAllmethod for User:

[Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findAll()

This method returns all instances of the User type.

Returns:

All User.

TIME: 40mins

Story 26:

findAllmethod for Bodycare product

[Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findAll ()

This method returns all instances of the Bodycare Product.

Returns:

All Bodycare Products.

TIME: 40mins

Story 27:

findAllmethod for Skincare product

[Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findAll ()

This method returns all instances of the Skincare Product.

Returns:

All Skincare Products.

TIME: 40mins

Story 28:

findAllmethod for Haircare Product

[Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findAll ()

This method returns all instances of the Product type.

Returns:

All Haircare Products.

TIME: 40mins

Story 29:

FindAllmethod for Medicine Product.

[Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findAll ()

This method returns all instances of the Product type.

Returns:

All Medicine Products.

TIME: 40mins

Story 30:

FindAllmethod for Babycare Product.

[Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<[T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> findAll ()

This method returns all instances of the Babycare Product type.

Returns:

All Babycare Products.

TIME: 40mins

Story 31:

existsById method for User:

Boolean existsById ([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) UserId)

This method returns whether an entity with the given UserId exists.

Parameters:

UserId - must not be null.

Returns:

True if an User with the given UserId exists, false otherwise.

Throws:

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if UserId is null.

TIME: 40mins

Story 32:

existsById method for Bodycare product

Boolean existsById ([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) BodycareId)

This method returns whether an entity with the given BodycareId exists.

Parameters:

BodycareId - must not be null.

Returns:

True if an entity with the given BodycareId exists, false otherwise.

Throws:

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if BodycareId is null.

TIME: 40mins

Story 33:

existsById method for Skincare product

boolean existsById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) SkincareId)

This method returns whether an entity with the given SkincareId exists.

Parameters:

SkincareId - must not be null.

Returns:

True if an entity with the given SkincareId exists, false otherwise.

Throws:

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if SkincareId is null.

TIME: 40mins

Story 34:

existsById method for medicine

Boolean existsById ([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) MedicineId)

This method returns whether an entity with the given MedicineId exists.

Parameters:

MedicineId - must not be null.

Returns:

True if an entity with the given MedicineId exists, false otherwise.

Throws:

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if MedicineId is null.

TIME: 40mins

Story 35:

existsById method for Babycare Product:

Boolean existsById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) BabycareId)

This method returns whether an entity with the given BabycareId exists.

Parameters:

BabycareId - must not be null.

Returns:

True if an entity with the given BabycareId exists, false otherwise.

Throws:

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if BabycareId is null.

TIME: 40mins

Story 36:

ExistsById method for Haircare Product.

Boolean existsById ([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) Haircare)

This method returns whether an entity with the given Haircare exists.

Parameters:

Haircare - must not be null.

Returns:

True if an entity with the given Haircare exists, false otherwise.

Throws:

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - if Haircare is null.

TIME: 40mins

Story 37:

Count method for User.

Long count ()

This method returns the total number of User available in the database.

Returns:

The total number of User.

TIME: 40mins

Story 38:

Count method for Bodycare Product.

long count()

This method returns the total number of Bodycare Products available in the database.

Returns:

The total number of Bodycare Products.

TIME: 40mins

Story 39:

Count for Haircare Product

long count()

This method returns the total number of Haircare Products available in the database.

Returns:

The total number of Haircare Products.

TIME: 40mins

Story 40:

Count for Skincare product

long count()

This method returns the total number of Skincare Products available in the database.

Returns:

The total number of Skincare Products.

TIME: 40mins

Story 41:

Count for medicine

long count()

This method returns the total number of Medicine available in the database.

Returns:

The total number of Medicine Products.

TIME: 40mins

Story 42:

Count method for Babycare Product.

long count()

This method returns the total number of Babycare Product available in the database.

Returns:

The total number of Babycare Product.

TIME: 40mins

Story 43:

deleteById method for User:

void deleteById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) UserId)

This method deletes a particular User with the given UserId, from the database.

Parameters:

UserId - must not be null.

Throws-

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given UserId is null.

TIME: 40mins

Story 44:

deleteById method for Bodycare Product.

void deleteById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) BodycareId)

This method deletes a particular Bodycare Product with the given BodycareId, from the

database.

Parameters:

BodycareId - must not be null.

Throws-

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given BodycareId is null.

TIME: 40mins

Story 45:

deleteById method for Skincare product

void deleteById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) SkincareId)

This method deletes a particular Skincare Product with the given SkincareId from the database.

Parameters:

SkincareId - must not be null.

Throws-

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given SkincareId is null.

TIME: 40mins

Story 46:

deleteById method for Haircare Product

void deleteById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) HaircareId)

This method deletes a particular Haircare Product with the given HaircareId, from the database.

Parameters:

HaircareId - must not be null.

Throws-

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given HaircareId is null.

TIME: 40mins

Story 47:

deleteById method for Medicine :

void deleteById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) MedicineId)

This method deletes a particular  Medicine with the given  MedicineId from the database.

Parameters:

 MedicineId - must not be null.

Throws-

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given  MedicineId is null.

TIME: 40mins

Story 48:

deleteById method for BabycareId Product

void deleteById([ID](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)  BabycareId)

This method deletes a particular Babycare Product with the given BabycareId from the database.

Parameters:

BabycareId - must not be null.

Throws-

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given BabycareId is null.

TIME: 40mins

Story 49:

delete method for User.

void delete([T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) User)

This method deletes a given User from the database.

Parameters:

User

Throws-

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given User is null.

TIME: 40mins

Story 50:

delete method for Skincare Product:

void delete([T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) Skincare)

This method deletes a given Skincare Product from the database.

Parameters:

Skincare

Throws-

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given Skincare is null.

TIME: 40mins

Story 51:

delete method for Bodycare Product.

void delete([T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) Bodycare)

This method deletes a given Bodycare Product from the database.

Parameters:

Bodycare

Throws-

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given Bodycare is null.

TIME: 40mins

Story 52:

delete method for Haircare Product.

void delete([T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) Haircare)

This method deletes a given Haircare Product from the database.

Parameters:

Haircare

Throws-

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given Haircare is null.

TIME: 40mins

Story 53:

delete method for Medicine:

void delete([T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) Medicine)

This method deletes a given Medicine from the database.

Parameters:

Medicine

Throws-

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given Medicine is null.

TIME: 40mins

Story 54:

delete method for Babycare Product.

void delete([T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html) Babycare)

This method deletes a given Babycare Product from the database.

Parameters:

Babycare

Throws-

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given Babycare is null.

TIME: 40mins

Story 55:

deleteAll method for User:

void deleteAll([Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<? extends [T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> User)

This method deletes the given User.

Parameters:

User -

Throws:

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given [Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true) is null.

TIME: 40mins

Story 56:

deleteAll method for Babycare Product

void deleteAll ([Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<? extends [T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> Bodycare)

This method deletes the given  Bodycare product.

Parameters:

 Bodycare

Throws:

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given [Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true) is null.

TIME: 40mins

Story 57:

deleteAll method for Haircare Product.

void deleteAll([Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<? extends [T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)>Haircare)

This method deletes the given Haircare Product.

Parameters:

Haircare

Throws:

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given [Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true) is null.

TIME: 40mins

Story 58:

deleteAll method for Skincare Product.

void deleteAll([Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<? extends [T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> Skincare)

This method deletes the given Skincare Product.

Parameters:

Skincare

Throws:

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given [Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true) is null.

TIME: 40mins

Story 59:

deleteAll method for medicine

void deleteAll([Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<? extends [T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> Medicine)

This method deletes the given entities.

Parameters:

Medicine

Throws:

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given [Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true) is null.

TIME: 40mins

Story 60:

deleteAll method for Babycare Product

void deleteAll([Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true)<? extends [T](https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html)> Babycare)

This method deletes the given entities.

Parameters:

Babycare

Throws:

[IllegalArgumentException](https://docs.oracle.com/javase/8/docs/api/java/lang/IllegalArgumentException.html?is-external=true) - in case the given [Iterable](https://docs.oracle.com/javase/8/docs/api/java/lang/Iterable.html?is-external=true) is null.

TIME: 40mins

Story 61: Spring Security

Adding spring security to our project.Spring Security is a framework that focuses on providing both authentication and authorization to java applications. It gives protection against attacks like session fixation, clickjacking, cross site request forgery, etc.

TIME: 5 hours.

Story 62: Mockito with Junit

Adding mockito with Junit in our project.

Mockito is a mocking framework, Java-based library that is used for effective unit testing of JAVA applications. Mockito is used for effective unit testing of Java Application. Mockito is used to mock interfaces so that a dummy functionality can be added to a mock interface that can be used in unit testing.

Create test cases for all methods in all classes.

**TIME**: 5 hours