JPA Auditing

Helps in recording the changes made to the entity. It keeps the track of who made the change and when it was made I.e. keeps the auditing trail.

We can add LocalDateTime createDate and LocalDateTime lastModified to the entity.

We can add @CreatedDate and @LastModifiedDate annotations to the fields.

But to enable jpa auditing we have to put @EnableJpaAuditing over the spring main class.
And also over the entity class we need to have @EntityListeners which takes up list of auditing listeners.

We have built in AuditingEntityListener.class.

@EntityListeners(AuditingEntityListener.class)

But now what happens that when we update the entity, the createdDate becomes null.

We can get rid of it by adding
@Column(nullable =false, updatable=false) on the createdDate field.

Similarly for the LastModifiedDate we can have
@Column(insertable=false), so it is only updatable, so when we insert entity it will be null and updatable by default has value true.

Now to take care of who had made changes:
Make a new class ApplicationAuditAware implements AuditorWare

AuditWare is generic by default which takes up what we have to track
If we are tracking only id

AuditorWare<Integer>

We have to override the Optional<Integer> getCurrentAuditor()

```
public Optional<Integer> getCurrentAuditor() {
      Authentication authentication =
SecurityContextHolder.getContext().getAuthentication();
      if (authentication == null || !authentication.isAuthenticated || authentication
instanceOf AnonymousAuthenticationToken) {
            return Optional.empty();
      }
```

```
        User user = authentication.getPrincipal();
        return Optional.nullable(user.getId());
}
```

```
// AnonymousAuthenticationToken means user is anonymous or cannot get the connection
// User is type of UserDetails
```

Create a bean of it.
Now we have to tell spring about it.

Put it in @EnableJpaAuditing(auditAwareRef="auditorAware")

Now we can add more instance variables in the entity:

```
@CreatedBy
@Column(nullable =false, updatable=false)
private Integer createdBy;

@LastModifiedBy
@Column(inertable=false)
private Integer lastModifiedBy;
```

We have to also specify

```
@Configuration
public class AuditorConfig {
   @Bean
   public AuditorAware<String> auditorProvider() {
      return new ApplicationAuditAware();
   }
}
```