<https://www.img2go.com/convert-video-to-gif>

Functional   
  confluence   
  jira   
  
Technical   
  Coding   
  
Engineering   
  branching stratergy   
  jenkins   
  sonar   
  
NFR   
  junit

Presentation Lead -

    Presentation Agenda   
  
            - Introduction to TMS   
              - link Confluence   
            - User Stories  
               - for Current Sprint   
               - Jira   
            - Team Introduction   
              Pod 1   
              - Niharika (Speaker)  
              - Sathiya  
              - Prasanth  
              - Hritikesh   
  
            - Sequence, Use Case, Class Diagram from each team on Confluence   
            - Data model in json format   
              trainer  
  
            {  
              trainerId: Integer,   
              trainerName: {  
                firstName: String,   
                lastName: String  
              }  
            }  
              - DB Structure Normalized   
            - Performing CRUD Operations with JDBC / Mongo   
            - Engineering Practices       
              - Showing branching stratergies   
              - Deploying application with Jenkins

ASDE Conceptual Problem statement   
--------------------------------------------------------------  
Day 1 to Day 9 – Lets do a CLI application  
  Day 1 to Day 5 - Deliverables -   
  
            Introduction to TMS   
            User Stories - for Current Sprint   
            Team Introduction   
            Sequence, Use Case, Class Diagram from each team on Confluence   
            Data model in json format   
              trainer  
  
            {  
              trainerId: Integer,   
              trainerName: {  
                firstName: String,   
                lastName: String  
              }  
            }  
            Performing CRUD Operations  with JDBC  
            Deploying application with Jenkins   
            Showing branching stratergies   
  
  
  
  Day 6 to Day 9 -   
            - Team Introduction   
              - User Stories - for Current Sprint (just put 1 more)  
            - Doing best practices with Java 11 features please not we are not using Java 1.8   
              + showcase SOLID Principles   
            - Code coverage to be shown with sonarqube if needed pls get access from sapient team   
              as they have cloud sonarqube   
            - Deploying application with Jenkins   
  
            Presentation ( Day 11)  
  
Day 10 to Day 12 – pax are expected to build the front end either with spring MVC or with Servlets + JSP  
Day 13 to Day 14 – build spring boot application for the core services  
Day 15 to 20 – consume end points with react (preferably lets use hooks)

Day 8 Revisit

* TDD – Test Driven Development ( Write test code first, then application code) (Test Driven Refactoring), Types of Testing (Unit, Integr, e2e, Load/Stress/Performance)
* Junit 5 (Annotations used @Test, @Before, @Assert, …. Why Testing – Cost Involved in removing bug in various env)
* Mockito (For creating mock objects for test purpose, Mock/Dummy/Fake/TestDouble. Spy, when, @Mock, @MockBean)
* Lombok (Java based library helps to avoid writing boiler plate code - @Getters, @Setters, @Data, @NoArgsConstructor, @AllArgsConstructor, @Log, @Slf4j, …)
* Logging/Logger – What & why. Logging approach. Java Logging, Log4j, Slf4J
* Logging Level – INFO,DEBUG, WARN, ERROR
* Can be redirected to console and/or file.
* SonarQube - Static Code Analysis Tool (SonarQube standalone server, SonarLint (IDE plugin), SonarCloud)

Day 9 Agenda

* Design Patterns
* SOLID Principles
* DRY
* YAGNI
* NFR

Design Pattern – Selecting a Optimal/Best solution to resolve a particular challenge in software development (IT industry)

Design Pattern – Effective way of solving challenges.

Writing Clean/Optimized Code

* Adding Sufficient comments (Documentation &/ Single line comments)
* Using Proper naming convention
* Proper Class/Interface/Variable name
* Package name should be in lowercase (reverse of company url)

Types of Design Patterns

* Creational (Factory, DI, Builder, Object Pool, Singleton, IoC)
* Structural (Adaptor, Decorator, Façade, Proxy)
* Behavioural (Chain Of Responsibility, Iterator, State, Visitor)
* Concurrency (Lock, Events, Thread Pool, Read Write Lock, Scheduler)

 SOLID Principles

S – Single Responsibility (SRP)

O – Open/Closed (OCP)

L – Liskov Substitution (LSP)

I – Interface Segregation (ISP)

D – Dependency Inversion (DIP)

Normalization in DB

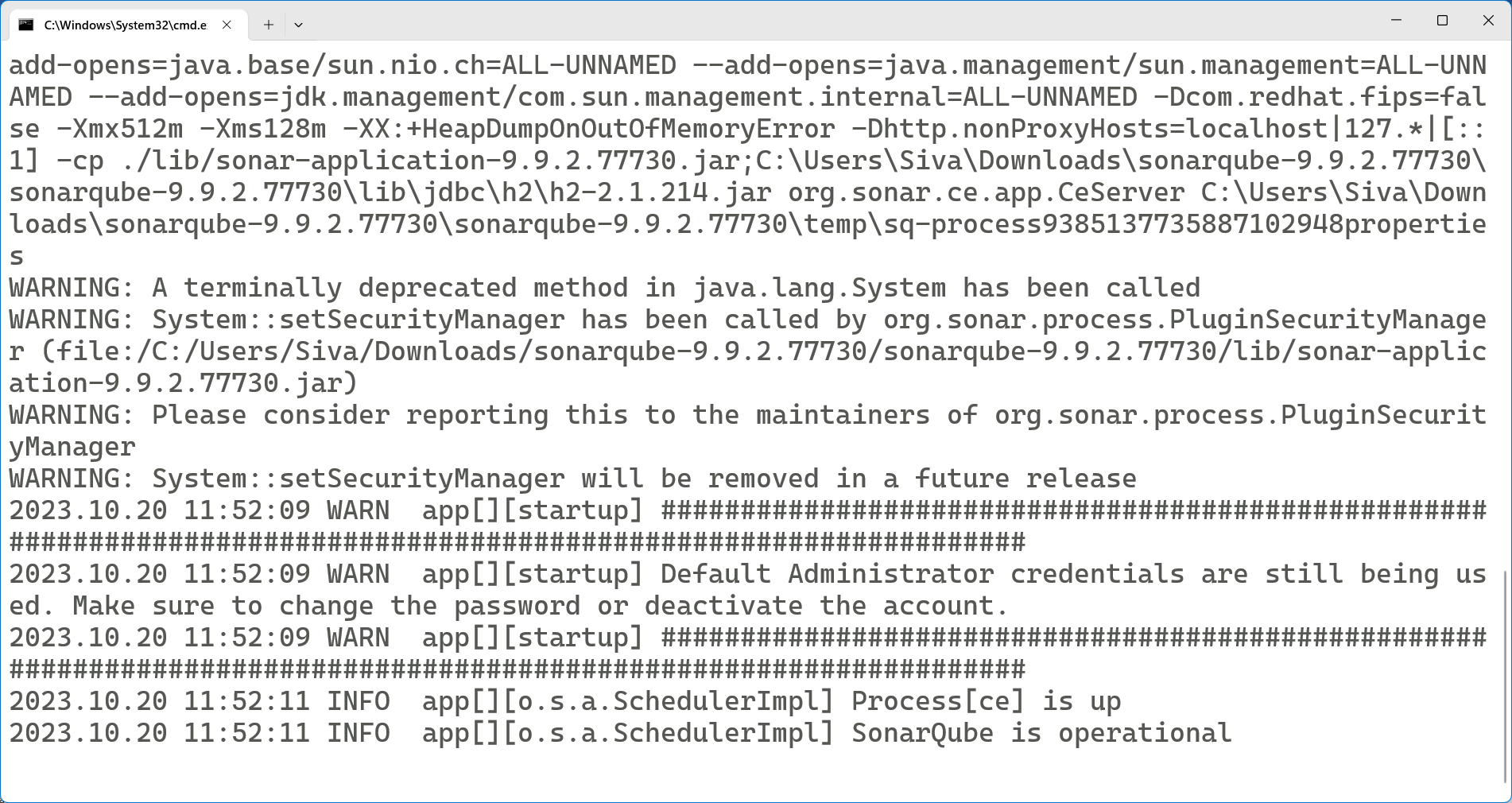
DRY – Don’t Repeat Yourself <https://www.baeldung.com/cs/dry-software-design-principle>

NFR – Non Functional Requirement (Performance) (Easy to Use [user friendly]

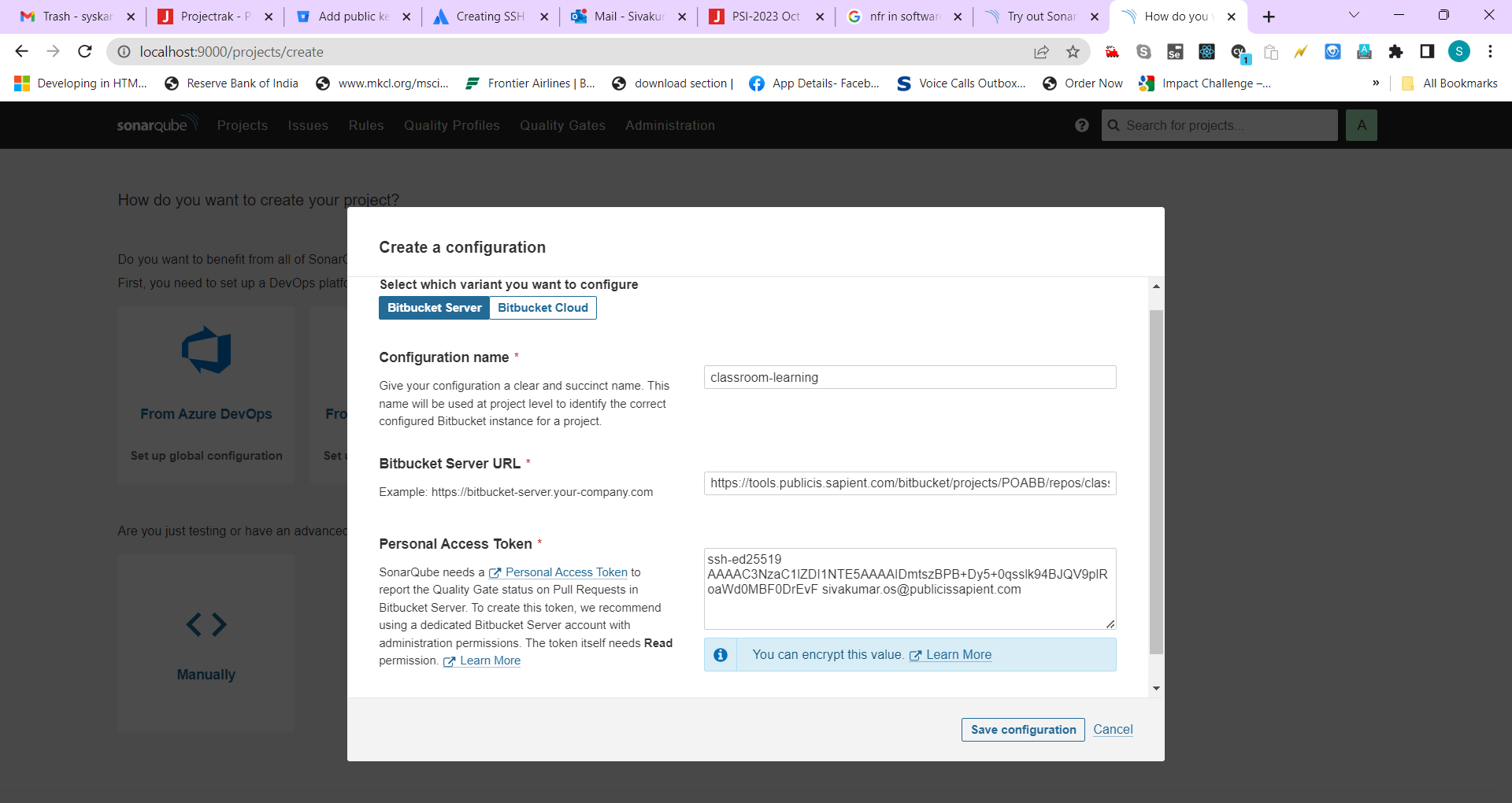
1. Scalability
2. Availability
3. Security
4. Configurability
5. Monitoring
6. Auditing
7. Integrating
8. Performance

SonarQube Demo

1. Download and install Sonaqube 9.9.2 (LTS) from official site
2. Make Sure JDK17 is available in the system & JAVA\_HOME env variable is added
3. Extract and open the bin folder, run StartSonar.bat file from windows sub-folder
4. In case of access-denied issues, tun the batch file as Admin (Run as Administrator)



1. Open browser and enter <http://localhost:9000>
2. Enter default username and password (admin/admin)
3. After first login, change the default password (admin123)



Create HTTP access Token in bitbucket – Profile 🡪 ManageAccount 🡪 Genereate HTTP Access Token (NzkyNjc3MDEwOTc1OvVH+UPNXG1tZqe+vb2x0oxGhAXC)

