# Checklist for Advanced Spring Boot Template Application

## Security and Authentication

#### **Security and Authentication**

* **JWT Authentication**
  + Implement Spring Security with JWT tokens.
* **Role-Based Access Control (RBAC)**
  + Configure roles and access control for different endpoints.
* **CSRF Protection**
  + Enable(for stateful web application like forms)/disable(for stateless APIs secured with JWT tokens) CSRF protection where necessary.
  + **Stateful Applications**: Enable CSRF protection to safeguard against CSRF attacks. Ensure CSRF tokens are included in forms
  + **Stateless Applications**: Disable CSRF protection when using JWT for APIs since CSRF protection is typically unnecessary for stateless REST APIs

#### **API Documentation**

* **Swagger/OpenAPI**
  + Configure Swagger for API documentation.

#### **Database Integration**

* **JPA/Hibernate**
  + Set up JPA for ORM and database interaction.
* **Database Migrations**
  + Implement Flyway or Liquibase for managing database schema changes.

#### **Exception Handling**

* **Global Exception Handling**
  + Use @ControllerAdvice to handle exceptions globally.

#### **Logging and Monitoring**

* **Spring Boot Actuator**
  + Set up Actuator for application monitoring.
* **Logging Framework**
  + Configure SLF4J with Log4j2 for logging.

#### **Testing**

* **Unit Testing**
  + Use JUnit 5 and Mockito for unit testing.
* **Integration Testing**
  + Use Spring Boot's test support for integration testing.
* **API Testing**
  + Use RestAssured for API testing.

#### **Configuration Management**

* **External Configuration**
  + Manage configuration with application.properties or application.yml.
* **Profiles**
  + Set up different profiles for dev, test, and prod environments.

#### **Asynchronous Processing**

* **Spring Async**
  + Implement asynchronous task execution with @Async.-Not required for the existing controllers but easy to implement.
* **Scheduling**
  + Set up scheduled tasks with @Scheduled. - Not required for the existing controllers but easy to implement whenever required.

#### **Caching**

* **Spring Cache**
  + Configure caching with Spring Cache abstraction.
* **Cache Providers**
  + Integrate with cache providers like Ehcache or Redis.

#### **Messaging**

* **Spring AMQP**
  + Integrate with RabbitMQ using Spring AMQP.
* **Kafka**
  + Integrate with Apache Kafka or Apache Flume whichever is better.

#### **File Handling**

* **File Upload/Download**
  + Implement file upload and download functionality.
* **Storage Integration**
  + Integrate with AWS S3 or similar storage services.

#### **Email Integration**

* **Spring Email**
  + Set up email sending with Spring Email.

#### **WebSocket Support**

* **Spring WebSocket**
  + Implement real-time communication with WebSockets: No need to implement in current project as it doesn’t fit the use cases of web socket. For more information on web socket refer : [Link](https://www.youtube.com/watch?v=TywlS9iAZCM)

#### **Correlation Id header:**

* **Correlation id**
  + Add co-relation id for all the APIs as a header.

#### **Internationalization (i18n)**

* **Localization**
  + Support multiple languages with i18n.