**Test cases**

**Audit-checklist service**

AuditChecklistApplicationTest (application)

Case 1: create mock for class and check not null using assertNotNull

QuestionsEntityTest (model)

Case 1: create mock for class and check the parameterized constructer

Case 2: check the questionId property

Case 3: check the auditType property

Case 4: check the questions property

Case 5: check the response property

AuditChecklistControllerTest (controller)

Case 1: create mock for all dependent classes

Case 2: check not null using assertNotNull

Case 3: check getQuestions method in controller with valid token

Case 4: check getQuestions method in controller with Invalid token

Case 5: check saveResponse method in controller with valid token

Case 6: check saveResponse method in controller with Invalid token

AuditTypeTest (pojo)

Case 1: create mock for auditType class

Case 2: check not null using setter and getter methods

Case 3: check constructors and toString methods

AuthResponseTest (pojo )

Case 1: create mock for authResponse class

Case 2: check constructors and toString methods

Case 3: check setter and getter methods

CustomErrorResponseTest (pojo)

Case 1: check not null for the object with default constructor

Case 2: check not null for the object with parameterized constructor

Case 3: check toString and all other property getter and setters

ChecklistServiceTest (service)

Case 1: create the mock for cheklistrepository and checklist service

Case 2: check the getQuestions without any exception

Case 3: check the getQuestions with exception

Case 4: check the saveResponse method

TokenServiceTest(service)

Case 1: create the mock for auth client ,token service,response

Case 2: check the token validity with valid token

Case 3: check the token validity with null pointer exception

Case 4: check the token validity with Invalid token

FeignProxyException(exception)

Case 1: create the mock for feignProxyException class

Case 2: check null pointer and constructor methods

GlobalErrorHandlerException(exception)

Case 1: create the mock for globalErrorException class

Case 2: check the not null and constructor methods

TokenExpiredException(exception)

Case 1: create the mock for tokenExpiresException

Case 2: check the not null and constructor methods

**Audit BenchMark Service**

AuditbenchmarkApplicationTest

Case 1: create mock for the application class

Case 2: check the not null for the mock

BenckMarkControllerTest(Controller)

Case 1: create mock for required classes like Authclient, Tokenservice, Benchmark, Benchmarkcontroller.

Case 1: check the benchmark

Case 2: check whether the token given is invalid.

FeignExceptionTest (exception)

Case 1: create mock for required classes like FeignProxyException.

Case 2: check Feign Constructor

GlobalExceptionHandlerTest (exception)

Case 1: create mock for required classes like GlobalExceptionHandler.

Case 2: check whether it handles wrong date exception format exception.

BenchmarkTest (model)

Case 1: Create a method with @Before to initialize object before test method calls.

``` Case 1: check the property auditType by set and get methods.

```Case 2: check the property noOfNos by set and get methods.

AuthResponseTest (pojo)

Case 1: check constructor

Case 2: check the property uid(user id).

Case 3: check the property isValid.

CustomErrorResponseTest(pojo)

Case 1: Create a method with @Before to initialize object before test method calls.

Case 2: Create a method with @After to initialize object after test method calls.

Case 3: check for parameterized constructor.

Case 4: check the property getStatus

Case 5: check the property setStatus.

Case 6: check the property setMessage.

Case 7: check the property getMessage.

Case 8: check the property setLocalDateTime.

BenchMarkService (service)

Case 1: Create mock for required classes like authclient, authResponse, tokenService.

Case 1: check for token validity when the token is valid.

Case 2: check for token validity when the token is Invalid.

Case 3: check for token validity when it gives null pointer Exception.