Deloitte Assessment for Java Developer:

# Approach :

## Selection of Framework:

1. Spring framework as an application framework for developing Java application has been meticulously selected for this Todo list implementation as Spring as a framework dates back to the year 2002 i.e it’s almost 2 decades old framework, with various enhancements on its way, which can be considered trustworthy in terms of built-in security features.
2. Majorly, Todo list implementation required faster development and continuous integration and testing of API’s developed as the time frame for this assessment was limited, Spring majorly focuses on speed, simplicity and productivity.
3. Todo list application involved implementation of RESTFUL web services for implementing basic CRUD-operations for Inserting/Deleting/Updating Todo list, below is the list of web services and their respective functionality.
4. Spring framework gives an added advantage of focusing on business logic more other than unnecessary configuration part, which made the task of writing web services easy, it also provides seamless integration for the development of restful web services

* /createItem : Allows user to create a Todo task after login.
* /GetUserAllItems : Fetch users Todo tasks as per the request.
* /editUserItem : Allows to edit the user tasks as per his/her preference.
* /deleteItem : Allows users to delete the task when his/her task is complete

The above API’s are developed keeping in mind the basic functionality of the TodoList.

* The approach involved the development of backend(web services) initially, then testing the required functionality through POST-MAN API and building the front-end over the APIs.
* I followed this approach of developing web services first because it made testing easy through POSTMAN
* This approach helped me maintain modularity within the code and allowed changes in my Database model, with changes as they appeared on the go.
* Web services for different functionality were implemented separately, which allowed the addition of different features without disturbing the existing code and structure.

# Design Pattern:

## MVC:

1. MVC design pattern in this application is majorly used to separate the various designing aspects of the application like Front-end(View), Backend(Model) and Web services(Controllers).
2. This separation of logic has helped to implement modularity in the code and achieve higher readability for better understanding.
3. In this project implementation model majorly involved backend database tables definitions and the getters and setter methods.
4. Controller involved majorly business logic(Web Services) to authenticate the user based on their email address and passwords and allowing the user to Create/Edit and delete their TodoList as per their choices. All these functionalities are implemented based on RESTFUL web services.
5. Spring MVC in specific creates a distinct division between JavaBeans, Models, Controllers and views which helps to configure each of the components securely and efficiently, which was needed in this assignment.
6. Well defined interface to the business layer is achieved in this application by creating DAO, Service and Util classes specifically by using Spring MVC Design pattern.
7. MVC design pattern has offered seamless integration with View technologies like Angular JS framework serving frontend.

# Proposed Enhancement:

## Email generation :

1. Email template through Gmail can be triggered on successful creation of a ToDo task informing the user about the date of creation and date of expiry for the task to do.
2. If the task is a group task, functionality of adding/tagging a Friend/Colleague in the task can be configured along with email functionality.

# Reasons for Delay/Problems Encountered:

## 1St attempt with HSQL DB :

1. As HSQLdb is an in-memory database, I faced an issue with the data not to be persistent after Tomcat Startup for testing, i.e I lost all the data entered previously and had to generate the data all over again.
2. I tried to configure the HSQLdb to store my data in the file format, which is an option specified in the HSQLdb documentation through changes made in applications.properties file.
3. This configuration changes for file-format storage for HSQLdb took time, but could not be achieved.
4. Finally, I decided to move to another in-memory database which is an H2 database and could configure it to store my data in file-format and could achieve persistency.
5. I would have researched the in-memory database in advance before I started developing my application to avoid the delay.
6. The major difficulty I faced while developing this application was, maintaining the session for the logged-in user through session token, later I could solve these issues by use of UUID(Universally Unique Identifier) supported by Java’s Spring Framework.